

Automated Brain Lesion Segmentation in Magnetic Resonance Images

Inauguraldissertation

zur
Erlangung der Würde eines

Dr. sc. med.

vorgelegt der Medizinischen Fakultät der Universität Basel

von
Simon Andermatt
aus Baar, Kanton Zug

Basel, 2019

Originaldokument gespeichert auf dem Dokumentenserver der Universität Basel

Original document stored on the publication server of the University of Basel

edoc.unibas.ch

Genehmigt von der Medizinischen Fakultät auf Antrag von
Prof. Dr. Philippe C. Cattin, Universität Basel – *Dissertationsbetreuer und Fakultätsverantwortlicher*
Dr. Jens Würfel, Universitätsspital Basel – *Korreferent*
Prof. Dr. Ender Konukoglu, ETH Zürich – *Externer Gutachter*

Basel, den 23. November 2018
Prof. Dr. Primo Schär, Dekan

Contents

<i>Acknowledgements</i>	vii
<i>Summary / Zusammenfassung</i>	ix
1 Introduction	1
2 Medical Background	3
2.1 Magnetic Resonance Imaging	3
2.2 Diseases of the Brain visible on MRI	6
3 Automated Lesion Segmentation	9
4 Artificial Neural Networks	19
4.1 Neural Networks	19
4.2 Recurrent Neural Networks	36
5 Multi-Dimensional Gated Recurrent Units	39
5.A Derivation of the Forward and Backpropagation	50
6 Supervised Lesion Segmentation using MD-GRU	55
7 Automatic Landmark Localization	69
8 Weakly-Supervised Pathology Segmentation	79
9 Automated Tracking of Lesions	89
10 Discussion and Conclusion	97
<i>References</i>	99
<i>Publications</i>	111

Acronyms

3D three dimensional. 5

AVD average (Hausdorff) distance. 15

BraTS brain tumor segmentation. 6, 12, 16, 17, 55

CNN convolutional neural network. 13, 17

CNS central nervous system. 1, 3, 6

CSF cerebrospinal fluid. 5, 12, 14

CT computed tomography. 3, 7

DNN deep neural network. 1, 16

DWI diffusion-weighted magnetic resonance imaging. 10

EM expectation maximization. 12, 14, 16

FCM fuzzy C-means. 12–14

FCN fully convolutional network. 16

FLAIR fluid attenuated inversion recovery. 5, 7, 9, 10, 13

FNN feed-forward neural network. 22

GAN generative adversarial network. 22, 98

GBM glioblastoma, also known as glioblastoma multiforme. 6, 7

GD gradient descent. 32

GM gray matter. 12, 14

GMM Gaussian mixture model. 12, 14

- GRU** gated recurrent unit. 22, 37, 38, 97
- HD** Hausdorff distance. 15
- ISBI** the International Symposium on Biomedical Imaging. 17
- ISLES** ischemic stroke lesion segmentation. 10, 12
- k-NN** k-nearest neighbors. 12
- KAP** Cohen kappa coefficient. 14, 15
- LMSLS** longitudinal MS lesion segmentation. 6, 10, 12, 55
- LSTM** long short-term memory. 22, 37–39
- MD-GRU** multi-dimensional gated recurrent unit. 2, 55, 69, 97, 98
- MICCAI** the International Conference on Medical Image Computing and Computer Assisted Intervention. 17, 39, 55, 79
- MPRAGE** magnetization-prepared rapid gradient-echo. 5
- MRF** Markov random field. 12–14, 16, 17
- MRI** magnetic resonance imaging. 2, 3, 6, 7, 9, 10, 12, 97
- MS** multiple sclerosis. 5, 6, 9, 10, 12, 14, 17, 55
- NN** artificial neural network. 13, 19, 30–32, 36
- PD** proton density. 5, 13
- RF** radiofrequency. 3, 4
- RNN** recurrent neural network. 19, 36–38
- SGD** stochastic gradient descent. 32
- VAE** variational autoencoder. 22
- WHO** World Health Organization. 6
- WM** white matter. 12, 14
- WMH** white matter hyperintensities. 9, 10, 17, 55

Acknowledgements

First and foremost, I want to thank Philippe Cattin for the amazing time at his group. His affirmative and supporting nature and open mind for unconventional ideas made this work actually possible. I am also indebted to my current and former colleagues at work: Simon Pezold, whom I could consult for nothing and anything, Antal Horváth, who helped me with any mathematical problem, Robin Sandkühler, who made working late enjoyable, Adrian Schneider and Ketut Fundana, for distracting me from work in hilarious and refreshing ways. Furthermore, I want to acknowledge the top technical support from Beat Fasel and Aydin Ürgen, with whom I was able to fix any possible technical issues and who handled any hardware request immediately. I am grateful to Ernst-Wilhelm Radue, Till Sprenger and Jens Würfel for establishing and maintaining the funding for my PhD studies through the MIAC AG. I am thankful for all the great collaborations I was allowed to take part in during my time at CIAN and want to especially thank Antal Horváth, Christoph Jud, Athina Papadopoulou, Simon Pezold, Orso Pusterla and Robin Sandkühler. I want to further thank all members of the CIAN/MIAC group, past and present, for fruitful discussions and making this PhD experience such a joyful adventure (in alphabetical order): Philippe Cattin, Natalia Chicherova, Corinne Eymann-Baier, Alina Giger, Antal Horváth, Christoph Jud, Nadia Möri, Peter von Niederhäusern, Simon Pezold, Frank Preiswerk, Tiziano Ronchetti, Robin Sandkühler, Adrian Schneider, Alex Seiler, Bruno Sempéré, Aydin Ürgen, Reinhard Wendler and Stephan Wyder.

Finally, I want to thank my family and friends for their generous support during my studies.

Summary / Zusammenfassung

Summary

In this thesis, we investigate the potential of automation in brain lesion segmentation in magnetic resonance images. We first develop a novel supervised method, which segments regions in magnetic resonance images using gated recurrent units, provided training data with pixel-wise annotations on what to segment is available. We improve on this method using the latest technical advances in the field of machine learning and insights on possible weaknesses of our method, and adapt it specifically for the task of lesion segmentation in the brain. We show the feasibility of our approach on multiple public benchmarks, consistently reaching positions at the top of the list of competing methods. Adapting our problem successfully to the problem of landmark localization, we show the generalizability of the approach. Moving away from large training cohorts with manual segmentations to data where it is only known that a certain pathology is present, we propose a weakly-supervised segmentation approach. Given a set of images with known pathology of a certain kind and a healthy reference set, our formulation can segment the difference of the two data distributions. Lastly, we show how information from already existing lesion maps can be extracted in a meaningful way by connecting lesions across time in longitudinal studies. We hence present a full tool set for the automated processing of lesions in magnetic resonance images.

Zusammenfassung

In dieser Dissertation wurde die automatische Läsionssegmentierung in Bildern der Magnetresonanztomografie (MRT) des Gehirns erforscht. Zunächst wurde mit Hilfe von überwachtem Lernen eine Methode entwickelt, welche Regionen auf MRT-Bildern mittels Gated Recurrent Units segmentiert, sofern Annotationen auf Pixelebene vorhanden sind. In Hinblick auf die Läsions-segmentierung im Gehirn wurde anschließend die Methode mit neuesten technischen Errungenschaften aus dem Forschungsgebiet des maschinellen Lernens und eigenen Erkenntnisse möglicher Schwachpunkte verbessert. An mehreren öffentlichen Datensätzen wurde gezeigt, dass die Methode konkurrenz-fähig ist. Anhand einer erfolgreichen Anwendung im Bereich der Landmarkenlokalisierung wurde die gute Generalisierbarkeit unserer Methode veranschaulicht. In einer weiteren Arbeit wurde die automatische Segmentierung im Bereich des

schwach-überwachten Lernens auf Datensätzen untersucht, für welche nur auf Bildebene Annotationen vorhanden sind. Basierend auf einem Datensatz von Patienten mit einer bestimmten bekannten Krankheit und einem gesunden Referenzdatensatz konnte die Differenz der zwei Datenverteilungen bestimmt und weitere, ungesehene Bilder von einem der beiden Datensätze segmentiert werden. Zum Schluss wird eine einfache Methode vorgestellt um Informationen individueller Läsionsentwicklung aus bestehenden, segmentierten Longitudinalstudien zu produzieren. Mit dieser Arbeit wird somit ein kompletter Satz an Methoden vorgestellt, welcher läsionsbehaftete Datensätze vollautomatisiert auswerten kann.

Chapter 1

Introduction

Motivation

Various different diseases affecting the central nervous system (CNS) cause some form of lesion in the tissue. For many diseases of the CNS, detection and quantification of such lesions is an important step towards disease diagnosis [81, 86, 88, 93, 94] and gives necessary insight on disease extent and progression, aiding substantially in planning an adequate treatment. Exorbitant amounts of volumetric lesion segmentations are required for medical drug trials as well as in medical practice to quantify or diagnose a specific disease. Those segmentations are drawn mostly by hand using sometimes semi-automatic techniques to help in the process. Manual segmentation is prone to subjective errors [105] and substantial inter- and intra-rater variability [37]. Furthermore, the exact quantification of such lesions through radiologists is a laborious, dull and time consuming task. A lot of money as well as valuable time of radiologists could hence be saved, were this task to be fully automated. Since the advent of deep neural networks for classification in [70], adopting them has created a sudden decrease in error metrics in various fields. Also, automated semantic segmentation has shown promising progress in recent years, with applications to natural images as well as medical data [26, 77, 97]. An application to lesion segmentation is therefore an obvious one. We want to explore different possibilities to gather information from the medical data in an automated fashion, without wasting human labor on the task, considering different scenarios. For tasks which have already been conducted by experts numerous times and produced a significant amount of training data, as well as for new tasks without training data, there is the need for automated means of solving them.

Contribution

When we started working on this project, deep neural network (DNN) were already applied to lesion segmentation [20]. In this thesis, however, instead of simply applying an existing DNN to the segmentation problem, we adapted *recurrent neural networks*,

an elegant form of recurrent computation on time series data, to segment anatomical structures, matching competing methods in accuracy. Treating each dimension along both direction once as temporal dimension, we can not only cut down on the number of weights, but also detect patterns with variations along one dimension without using a number of different filters for this task. We tuned our method to the problem of lesion segmentation, producing results beating the state of the art, which we confirmed on a number of public benchmark datasets. We showed the generalizability of our method by adapting it to a regression through classification problem, where we estimated a landmark coordinate in volumetric data. All these mentioned methods required lots of expensive manual training data, where for each new task, new training data had to be produced. We hence proposed a new method, which only requires a single binary image-level label stating if the image contains a certain pathology. Using only this information, we were able to produce results very close to the segmentations of fully supervised methods. Finally, we show a simple way of using already existing lesion maps from longitudinal studies to extract individual lesion development information. With this thesis, we provide a strong set of tools for the fully automatic segmentation of lesions in brain magnetic resonance imaging (MRI).

Outline

In Chapter 2, we will focus on the medical background of lesions in the brain. Chapter 3 gives a short overview on semantic segmentation and its application to brain lesion segmentation. Chapter 4 outlines different techniques of machine learning with neural networks and deep learning. We introduce the multi-dimensional gated recurrent unit (MD-GRU) in Chapter 5, and investigate its application to the task of volumetric brain anatomy segmentation. We tailor MD-GRU to the problem of lesion segmentation in Chapter 6, evaluating different modifications to our method. Chapter 7 underlines the flexibility of our formulation, applying MD-GRU to the problem of landmark detection. In Chapter 8, we move on to data without manual annotations, which have only been classified as healthy or pathological, and propose a formulation to train a pixel-wise segmentation algorithm using only this information. In Chapter 9, we show a method to quickly extract information on the temporal development of individual lesions in longitudinal studies, given lesion maps are already available. Finally, we conclude with a discussion of our results in Chapter 10.

Chapter 2

Medical Background

Injuries to the CNS can take various forms due to different causes, disease types and areas of injury, where we focus here on macroscopic lesions in the brain. A number of different imaging modalities exist to visualize such pathologies, largely depending on their characteristics. Since our area of interest is enclosed in the skull, we are restricted to non-invasive imaging modalities that can penetrate the skull, such as MRI and computed tomography (CT). CT has the advantage that images of high resolution can be produced but lacks soft tissue contrast, which is important for any CNS imaging. MRI features a relatively weak spatial resolution, since recording an image is a sequential process and time grows quadratic or cubic for 2D and 3D-imaging, respectively. Furthermore, long acquisitions are prone to movement artefacts, whereas shortening the acquisition time leads to a small signal to noise ratio. It is nevertheless the method of choice for most brain lesion imaging, due to it being free of harmful radiation and its remarkable soft tissue contrast.

2.1 Magnetic Resonance Imaging

MRI is made possible due to the collection of microscopic magnetic moments of hydrogen atoms in the tissue. When a strong magnetic field is applied, all these magnetic moments reorient themselves according to their position in the field. Using radiofrequency (RF) waves close to the so-called *Larmor frequency*, these magnetic moments can be excited. Following this event, the moments will slowly precess back to the direction dictated by the main magnetic field, which is called relaxation. During this precession, RF waves are emitted and can be recorded. Relaxation can be distinguished into T1 relaxation or longitudinal relaxation as well as T2 relaxation or transverse relaxation. T1 relaxation time is the time needed until the net magnetisation is at about 63% of its initial value. T2 relaxation time on the other hand is the time needed for the transverse component of the magnetisation relative to the main field to reach about 37% of its initial value [17]. The T1 and T2 times are properties inherent to the tissue and can be used to properly design their contribution to an image by choosing an adequate

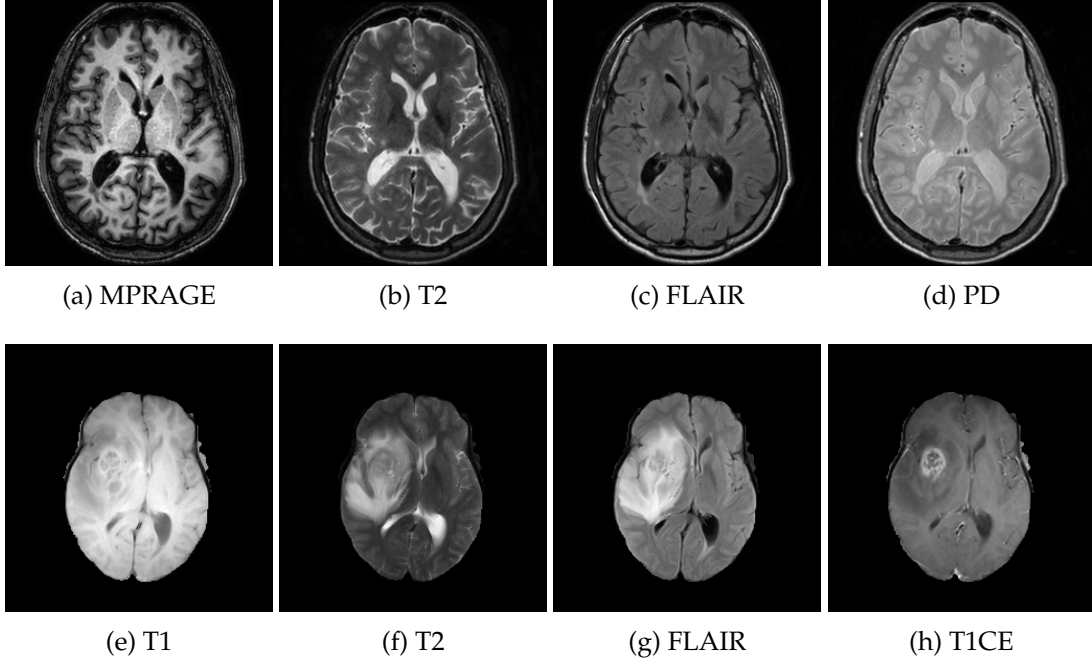


Figure 2.1: *Top row:* Slices from MPRAGE, T2, FLAIR and proton density (PD) weighted sequences from the longitudinal MS lesion segmentation challenge dataset (before pre-processing). *Bottom row:* Slices from T1, T2, FLAIR and T1 with applied contrast agent (T1CE) from the Brain tumor segmentation (BraTS) dataset (co-registered, interpolated to a resolution of 1 mm^3 and skull stripped).

imaging *sequence*.

Imaging Sequences

By selectively exciting different parts of the volume of interest, we can measure each voxel's individual response. Spatial coding of the signal is accomplished using three orthogonal gradient coils, which add a linear decay of field strength along their axis. Each of them either performs slice-selection, phase encoding or frequency encoding. Slice-selective excitation is used in 2d imaging, where applying a gradient results in only a slice of the volume being excited when transmitting the RF pulse, since only there the Larmor frequency matches the frequency of the RF pulse. With phase encoding, we can influence the phase of the magnetic moments in the volume linearly depending on their location along said axis. In 3d imaging, phase encoding is applied along two dimensions. Finally, with the frequency coil, we can directly encode the last dimension in the frequencies of the received signal by applying a gradient to the remaining axis. Contrary to the phase encoding, the frequency encoding does not influence the duration of a sequence. When scanning the whole so-called *k-space* by sampling all frequencies from each phase encoding, an inverse Fourier transform can recover the intensities at each

location in the slice or volume. Using the physical properties of magnetic moments as well as the gradients and main magnetic field of the scanner, a lot of different sequences can be applied, resulting in images highlighting different aspects of the tissue, as can be seen in Fig. 2.1.

There are various parameters that influence this process, with the echo and repetition time being two of the most important ones. Echo time denotes the time spent until the spin echo is read out using the frequency encoding gradient. The spin echo is the resonance in magnetic moments when their realignment induces a recordable signal and can be achieved using an inversion pulse at half the echo time. Repetition time is the time in between two subsequent excitations. To emphasize the T1 contribution, echo and repetition time are chosen short. Since, in contrast to water, fat quickly realigns its net magnetization to the main magnetic field, fat appears bright whereas water appears dark on T1 weighted images (see e.g. Fig. 2.1e). For T2 weighted images, both echo and repetition time are chosen longer. The resulting images show high intensities for both fat and water (see e.g. Figs. 2.1b and 2.1f). Lastly, by choosing a long repetition time and a short echo time, so called proton density (PD) images can be produced. By minimizing the difference in contribution of T1 and T2 time, the tissues with high concentration of protons produce the strongest signal, hence the name (see e.g. Fig. 2.1d). Furthermore, there are a number of general classes of acquisition techniques worth mentioning such as spin echo, gradient echo and inversion recovery. Spin echo sequences use a 90 degree and a 180 degree pulse, where gradient echo sequences use instead of the second pulse an inverse frequency gradient, resulting in a faster acquisition. Inversion recovery adds an additional 180 degree to the front of a spin echo sequence, which inverts the total magnetization. The 90 degree pulse is applied exactly at the point in time, called inversion time, where the longitudinal magnetization reaches zero of tissue we would like to suppress.

In the following, selected special sequences appearing in the remainder of this thesis, which are especially useful for the visualization of lesions, are quickly discussed.

3D-MPRAGE 3D magnetization-prepared rapid gradient-echo (MPRAGE) imaging [91] has been designed for a fast acquisition of T1 weighted high resolution scans. 3D MPRAGE has been shown to be superior in indicating focal lesions compared to traditional T1 spin echo sequences [19]. An exemplary slice of a resulting image is shown in Fig. 2.1a.

FLAIR Fluid attenuated inversion recovery (FLAIR) [29] is an inversion recovery sequence with a long inversion time which suppresses the cerebrospinal fluid (CSF) signal. This suppression makes imaging of lesions possible which are adjacent to the ventricles or the CSF in general and has been shown to be superior to T2 weighted images for the detection of multiple sclerosis (MS) lesions [14]. Examples of FLAIR images can be seen in Figs. 2.1c and 2.1g.

Contrast enhanced T1 Gadolinium, usually applied in conjunction with T1 sequences, drastically reduces the T1 time of surrounding tissue, resulting in a bright signal on T1 where it accumulates. It is used to demonstrate focal lesions such as tumors or active lesions in MS [35, 71, 87]. An example of a contrast enhanced T1 scan can be found in Fig. 2.1h, where the tumor core shows hyperintense tissue due to a higher contrast agent uptake.

2.2 Diseases of the Brain visible on MRI

There are many medical conditions leading to signs of deterioration in the brain which are visible on MRI. Trauma can lead to hemorrhages or swelling of tissue, resulting in visible lesions in the brain. MS leads to a number of focal lesions visible on different MRI sequences. Other examples for lesions in the brain are tumors or damaged tissue after a stroke. The reasons for lesions in the brain are manifold, and a complete characterization of all possible diseases and their appearances is outside of the scope of this thesis. Even though our methods could be used for many types of lesions in the brain, we concentrate for brevity on the pathologies we came into contact with during our investigations in this PhD project. In our studies, we use data from patients suffering from MS [22] and brain tumors in the form of glioblastoma (including lower grade glioma) [10–12, 86]. Figure 2.1 shows exemplary slices of the longitudinal MS lesion segmentation (LMSLS) challenge and the brain tumor segmentation (BraTS) challenge, respectively. In the following, we will shortly describe these diseases, their progression and implication as well as their appearance on different MRI sequences.

Multiple Sclerosis MS is a disorder of the CNS of presumably autoimmune nature [82]. MS is characterized by the degeneration of myelin sheaths, the insulation of neuronal axons, which hinders signal amplification and therefore results in signal loss. The disease is characterized by the formation of focal lesions as well as overall atrophy of the nervous tissue [113]. A number of MRI sequences can be used to visualize different features of focal MS lesions. T2 weighted scans can be used to quantify the total lesion load [39], where FLAIR has shown to be more sensitive and demonstrates a larger number of lesions [14] than T2. Gadolinium administered during acquisition of a T1 weighted scan visualizes active lesions. Lesions not appearing on such a scan, but that are hyperintense on T2 can be classified as chronic [66]. Persistent T1 hypointense lesions, so called black holes, are used as markers for axonal loss and neuronal tissue damage.

Glioblastoma and Lower Grade Glioma Astrocytomas are tumors originating from astrocytes, a special type of glial cell in the CNS. The World Health Organization (WHO) defines four grades for astrocytomas with increasing malignancy, where the fourth grade is represented by glioblastoma, also known as glioblastoma multiforme (GBM). GBM is the most common cancer which starts in the brain, comprising 15 % of intracra-

nial neoplasms (new and abnormal growth of tissue) and 60 to 75 % of all astrocytomas [127]. Diagnostic modalities for GBM include CT, MRI and histology. CT is primarily used for initial screening and MRI for further characterization, where a typical protocol consists of a T1, T2, FLAIR and T1 contrast enhanced sequence. Gliomas often show a contrast-enhanced ring with a hypointense core on contrast enhanced T1, and T2 and FLAIR can visualize the degree of edema around the glioma [127].

Chapter 3

Automated Lesion Segmentation

Image segmentation is the task of grouping neighboring pixels or voxels in images to meaningful segments. In medical image analysis, image segmentation usually refers to semantic segmentation, where each segment is also assigned a label and all areas with a given label share certain characteristics. This can be a meaningful separation of foreground and background or classification of each segment to a predefined class, such as different anatomical regions. Semantic segmentation is hence closely related to classification, as we assign a label or class to each voxel in the image. Hereafter we will use the term segmentation to refer to semantic segmentation if not explicitly stated otherwise.

In contrast to healthy tissue, lesions can in most cases take on arbitrary shapes and appear at different locations in the brain. Lesions in MS patients which are visible on clinical MRI are located primarily in the white matter, and can be elongated depending on the structure of tissue. They occur as heterogeneous spots and depending on their state can appear hyperintense on T2 weighted scans including FLAIR, and on T1 weighted scans when contrast agent has been administered. They can appear as hypointense “black holes” in T1-weighted scans, where sometimes more complex shapes and patterns can form in the case of confluent lesion types, e.g. a lesion with an active, enhanced part and a passive black hole part.

White matter hyperintensities (WMH) show similarities to MS lesions appearing on T2-weighted scans but contain slightly less sharp lesion boundaries [21].

The lesion shape is more arbitrary for brain tumors as compared to the previously discussed lesions, with less heterogeneous intensity inside the tumor region. Tumor lesions are usually larger, and affect one localized part in the brain. Tumor subtypes show distinct areas with different semantic meaning. In the case of gliomas, edemas surrounding the tumor can be seen as hyperintense diffuse structures on T2 and FLAIR images. The tumor core can be subdivided into a hyperintense part visible on contrast enhanced T1, a necrotic area with dark, hypointense regions on T1, usually in the center of the tumor and the remaining tumor tissue, which is visible on T2 as slightly hyperintense [44, 86].

Ischemic stroke lesions appear differently during their temporal development. First,

a lesion can be visualized as strongly hyperintense in diffusion-weighted magnetic resonance imaging (DWI) and moderately hyperintense in FLAIR. About two weeks later, the lesion will show more hyperintensity in FLAIR while being isointense in DWI. Edema can build up around the lesion and disappear again. Shape, location, size and even their number vary between patients. Furthermore, especially in older patients, a differential diagnosis to WMH might be difficult [79].

Due to the high variability of shapes and the usually unclear lesion boundaries, it is not an easy task to model the possible appearances of lesions in the brain, even when focusing on one disease only (see for example Fig. 3.1). Disease-independent factors further add to the difficulty of automatically segmenting lesions, since there is usually a large inter- and intrarater variability in manual delineations of lesions [36]. Especially for supervised segmentation methods, which take most or all information on how to correctly segment from manually labeled examples, this creates an upper bound in measurable accuracy. A related issue is the so called expert knowledge, which is highly process dependent. If an expert is taken out of her routine, or asked to dwell on a decision for some more time, the decision taken might significantly differ [105]. Hence, even though segmentation of lesions itself is a difficult task and requires a lot of domain knowledge in human experts, automated methods could help provide an objective means of quantification. In the following, we go through a selection of important works on the topic of brain lesion segmentation in MRI.

Brain Lesion Segmentation in the Literature

The body of research of general lesion segmentation in brain MRI is too large to be exhaustively covered in a thesis. Fortunately, Garcia et al. [38] and Lladó et al. [76] provide with their review papers valuable information for the segmentation of MS lesions prior to 2013. Gordillo et al. [44] adequately summarize progress on brain tumor segmentation until 2013 and Rekik et al. [96] on ischemic stroke until 2012. For the task of WMH segmentation, Caligiuri et al. [21] summarize WMH specific algorithms until 2015, albeit a lot of methods initially developed for MS are also applied for this task. Akkus et al. [1] and Havaei et al. [47] focus on the application of deep learning to lesion segmentation in 2016 and 2017 respectively. Since there is a large overlap of applicable methods between disease types, we decided to summarize findings from the above review papers in the following sections and complement the list with information from more recent methods which have been validated on a public benchmark or challenge. Analyzing methods from different public benchmarks allows us to close in on more recent, competitive methods which together define the state of the art in brain lesion segmentation. As above mentioned reviews cover the state before 2012 quite well for all methods of the most popular disease types for segmentation, we include in our investigation challenges from or after 2012. These include the brain tumor segmentation challenges of 2012–2016 [13, 34, 84–86], the LMSLS challenge from 2015 [22] and the cross-sectional MS lesion segmentation challenge from 2016, the WMH challenge from 2017 and the ischemic stroke lesion segmentation (ISLES) challenges from 2015 to

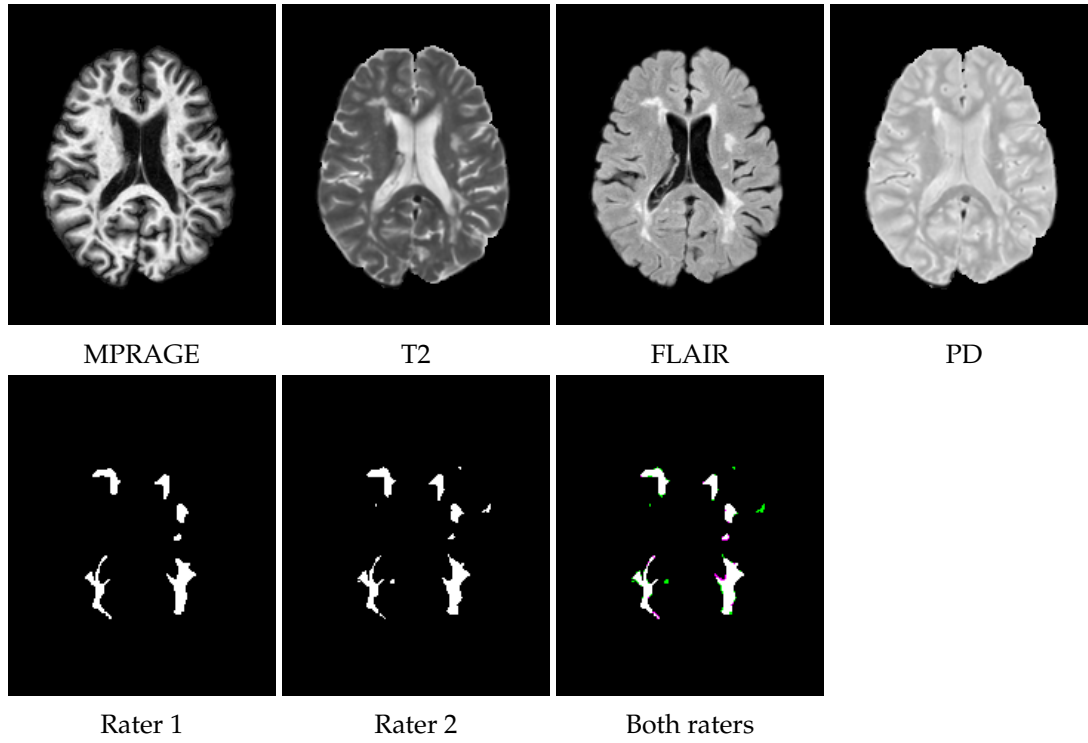


Figure 3.1: *Top row:* Slices from MPRAGE, T2, FLAIR and proton density (PD) weighted sequences from the longitudinal MS lesion segmentation challenge dataset (before pre-processing). *Bottom row:* respective binary segmentation into lesion and background by two raters and their agreement with areas segmented only by one rater color coded in green and magenta.

2017 [79]. Furthermore, there have been journal publications summarizing results of some of the challenges, including the BraTS challenges in 2012 and 2013 [86], the ISLES challenge from 2015 [79] and the LMSLS challenge in 2015 [22].

We structure this section as follows. First, we introduce common preprocessing steps. We then distinguish between supervised and unsupervised methods and address them in individual sections. Finally, we will introduce and discuss a number of performance measures and analyze the state of the art in the last two subsections.

Preprocessing

Garcia et al. [38] and Lladó et al. [76] list a number of common preprocessing steps for the segmentation of MS lesions, such as coregistration to the patient space or another reference space, which is especially important if an atlas is going to be used later on for segmentation. Furthermore, skull stripping or brain extraction methods can help reduce possible outliers. Intensity inhomogeneity correction is often used due to the inherent inhomogeneity of magnetic fields applied in MRI. Noise reduction can help to overcome negative effects of noise in the image. Depending on the assumptions that are made for a particular method, it can be useful to normalize the intensities to a predefined range. The preprocessing steps are mostly the same independent of disease types [1, 21].

Supervised Methods

Lladó et al. [76] group the supervised segmentation methods into atlas and manual segmentation based. The atlas based methods first register a statistical or topological atlas to the sample to be segmented. This atlas is in turn used as prior information to classify the pixels or voxels into different tissues and lesions can be segmented as outliers of the model. This model can be based on intensity values, using a clustering method such as k-nearest neighbors (k-NN) [126], expectation maximization (EM) of a Gaussian mixture model (GMM) [111], a fuzzy C-means (FCM) [107–109] and other methods [38, 76]. On top of the intensity information, such a model can also include neighborhood information, for instance through a Markov random field (MRF) [121] and other means [38, 76].

Other atlas-based methods estimate lesion probabilities directly using the tissue priors as additional input to a classifier [132]. The manual segmentation based approach requires data which have already been labeled by hand or through another automated method. After training the method on the labeled data, it can be used to segment unseen data. Early attempts use a variety of classifiers directly on the intensity information [76]. Instead of using the intensity directly, features can be defined to be used for training either alone or together with the original data. Such features include the white matter (WM), gray matter (GM) and CSF tissue probabilities produced by another model [2, 62], features derived from local thresholding maps and morphological properties of the resulting segments [42], spatial features [3–5], vector image joint

histograms built over feature vectors indicating lesions [106], or a large pool of features [90]. Other methods use derived features or properties from the labeled training data to train a parametric method, such as a MRF [102, 115] or a graph cut [72], or create an explicit model for healthy structure where lesions are detected as outliers [90].

Gordillo et al. [44] divide the approaches for glioma segmentation into supervised and unsupervised, but give a further distinction in threshold-based, region-based, pixel-based and model-based methods. Threshold-based and region-based approaches include a lot of semi-automated methods, which are not going to be covered here. The pixel-based subgroup contains methods using artificial neural networks (NNs), FCM and MRFs, whereas the latter two are usually applied in an unsupervised setting. Also the model-based subgroup is applied in the unsupervised setting, where active contours or level sets are iteratively adjusted to fit a predefined energy function [44]. Recent methods are increasingly based on deep learning, especially convolutional neural networks (CNNs). Havaei et al. [47] give a valuable overview of currently successful methods. They pay more attention to the task itself, describing how to prepare the data and details of the training procedure. They distinguish data processing in 2, 2.5 and 3 dimensions, where 2.5D is defined as processing the 3D volume independently from different directions in 2D. Amongst others, they list encoder decoder methods such as convolutional encoder networks [20] and multi-directional long short-term memory networks [114], the foundation of one of the methods in this thesis. Akkus et al. [1] divide the CNN based methods in three categories, the patch-wise, semantic-wise and cascaded CNN architectures. Patch-wise architectures classify each neighborhood of a pixel/voxel individually, while semantic-wise CNN architectures directly output the segmented patch, such as for instance the U-net [97] which uses a fully convolutional architecture. Cascaded CNN architectures finally are combinations of multiple CNNs, where the output of the first is used as input for the second network.

Unsupervised Methods

Unsupervised methods rely on unlabeled training data only, sometimes neither knowing how many classes to divide the data in nor knowing what meaning a given label has. Gordillo et al. [44] state that unsupervised segmentation is a narrow area of research for tumor segmentation, since it is hard to define shape priors or intensity priors, but list FCM and MRFs as popular unsupervised segmentation methods. Furthermore, a variety of self organizing maps have been combined with FCM [122]. Surveys on the performance of popular unsupervised methods in 2015 and 2016 claim almost comparable performance to supervised algorithms [61, 103]. In the following years, this statement was quashed through the introduction of supervised deep learning methods [46, 63, 65, 92].

For MS lesion segmentation Garcia-Lorenzo et al. [38] explain, that some methods take into account, that lesions can be modelled with their appearance in different sequences. They can be hyperintense on T2, PD or FLAIR and usually appear inside the normal appearing white matter. Using the fact that T1 provides good contrast for

anatomy segmentation, these methods model four individual classes (GM, WM, CSF and lesions). They usually apply FCM or GMM-EM for this task. Some methods model the lesion class implicitly as outliers of a normal appearing brain tissue model. Instead of only using intensity and atlas information per pixel, spatial information can be incorporated through methods such as MRFs. Other methods segment the image nonsemantically using parcellation algorithms such as watershed or mean shift with subsequent classification of subregions [38]. Llado et al. [76] differentiate between methods explicitly modelling tissue and either modelling lesions as additional class or as outliers of the tissue classes and methods that explicitly only segment lesions. The methods of the first group depend largely on the quality of the tissue segmentation step, whereas the methods from the latter group usually work for special lesions such as enhancing lesions, since a lot of parameters have to be tuned to the respective sequences by hand [76]. Sparse coding and dictionary learning has been proposed to detect irregular anatomy in an unsupervised fashion [124]. Unsupervised domain adaptation has been investigated to transfer knowledge from one domain to another for instance across different scanners [64]. Although segmentation accuracies comparable with supervised methods are attainable, this method still needs dense labels in the other domain. Jain et al. [59] introduce methods for cross-sectional and longitudinal MS studies, where in the former, lesions are segmented as outliers, similar to [121]. Additionally, unrealistic outliers are dismissed as tissue outliers [59]. The latter additionally incorporates temporal information from two subsequent scans [58].

Performance Measures

Popular performance measures for lesion segmentation can be grouped into pixel-wise and lesion-wise metrics. Pixel-wise metrics operate on pixel or voxel values directly while lesion-wise metrics use properties of individual clusters in the segmentation. Taha and Hanbury [117] define the following categories for popular segmentation metrics: spatial overlap based, volume based, pair counting based, information theoretic based, probabilistic and spatial distance based metrics.

Popular such measures include the Dice and Jaccard (Jac) indices, the true positive and true negative rates (TPR and TNR), the false positive and false negative rates (FPR and FNR) among others for the spatial overlap methods, volume similarity (VS) for the volume methods, mutual information (MI) for the information theoretic and the Cohen kappa coefficient (KAP) for the probabilistic cases. In the case of binary segmentation, methods from the first five categories can be characterized through a combination of the four cardinalities of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) clusters or pixels. We define the four cardinalities as follows for a binary reference label map or *ground truth* T and a given binary segmentation S , where

N is the total number of pixels or voxels:

$$\begin{aligned} TP &= \sum_i^N T(i) \odot S(i), & FP &= \sum_i^N S(i) - TP, \\ FN &= \sum_i^N T(i) - TP, & TN &= N - TP - FN - FP. \end{aligned}$$

Using these, we can quickly define the above mentioned spatial metrics:

$$\begin{aligned} \text{Dice} &= \frac{2TP}{2TP + FP + FN}, & \text{Jac} &= \frac{TP}{TP + FP + FN}, \\ \text{TPR} &= \frac{TP}{TP + FN}, & \text{TNR} &= \frac{TN}{TN + FP}, \\ \text{FPR} &= 1 - \text{TNR}, & \text{FNR} &= 1 - \text{TPR}. \end{aligned}$$

The VS and MI can be defined similarly using the cardinalities:

$$VS = 1 - \frac{|FN - FP|}{2TP + FP + FN}, \quad MI = H_m(TP + FP) + H_m(TP + FN) - H_j,$$

where $H_m(\cdot)$ and H_j are the marginal and joint entropies respectively:

$$\begin{aligned} H_m(X) &= -\frac{X}{N} \log \frac{X}{N} - \frac{N-X}{N} \log \frac{N-X}{N}, \\ H_j &= - \sum_{i \in \{FP, TP, FN, TN\}} \frac{i}{N} \log \frac{i}{N}. \end{aligned}$$

Finally, KAP is defined as follows:

$$KAP = \frac{f_a - f_c}{N - f_c},$$

where $f_a = TP + TN$ and $f_c = \frac{(TN+FN)(TN+FP) + (FP+TP)(FN+TP)}{N}$. It measures agreement by taking into account the possibility of agreement by chance. All these measures are quite resilient to outliers, since the spatial location of mislabeled pixels is not taken into account. Figure 3.2 shows a selection of measures as a function of false positive and true positive pixels, where the reference segmentation is fixed at 1/4th of an image of 100×100 pixels.

Spatial distance based metrics, such as the Hausdorff distance (HD) and to a lower extent also the average (Hausdorff) distance (AVD) can be quite sensitive to outliers in the segmentation. They are defined as follows using the directed HD $h(\cdot, \cdot)$ and the directed AVD $d(\cdot, \cdot)$:

$$\begin{aligned} HD &= \max(h(C_S, C_T), h(C_T, C_S)), & h(X, Y) &= \max_{x \in X} \min_{y \in Y} \|x - y\|, \\ AVD &= \max(d(C_S, C_T), d(C_T, C_S)), & d(X, Y) &= \frac{1}{N} \sum_{x \in X} \min_{y \in Y} \|x - y\|, \end{aligned}$$

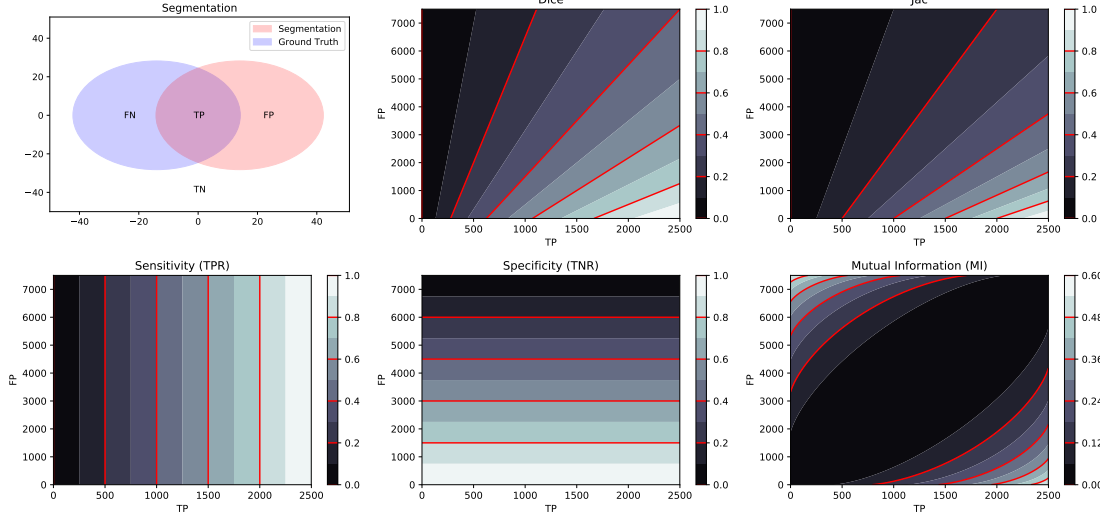


Figure 3.2: Popular segmentation metrics. *Top, left*: Segmentation task with annotated regions (F=False, T=True, P=Positive, N=Negative). *Top, right*: Dice and Jaccard indices as function of the segmentation (TP and FP). *Bottom*: true positive rate, true negative rate and mutual information as function of the segmentation (TP and FP).

where C_S and C_T are the set of coordinates of all pixels or voxels which are set to 1 in the segmentation maps S and T , respectively.

For further measures and generalizations to fuzzy segmentations or multi-class situations, Taha and Hanbury [117] provide a concise survey.

State of the Art

The BraTS challenge has been held each year since 2012 and is hence an invaluable indicator of recently popular methods in tumor segmentation. In the first two years, manual segmentations were used for supervision, while in the later years labels merged from the winning methods in the first two years replaced the manual labels. Only in 2017, manual labels were reintroduced again, final results for this competition are unfortunately not yet available. In 2012 and 2013, successful approaches for segmenting tumors were based on decision tree ensembles or random forests, MRF approaches on different features, cellular automata and EM segmentation [86]. None of the methods were based on DNNs. In the following years, deep learning based methods started to take ground in brain tumor segmentation, with 2 of 15 methods in 2014 [47] and 7 of 12 methods in 2015 [84]. Although the overall winner of 2015 was a semi-automated method, in both 2014 and 2015, the winning approaches amongst the fully automated methods were deep learning based. In 2016, a simple fully convolutional architecture made first place, and 9 of 16 fully automated methods were based on or included deep learning [85]. In 2017, an ensemble of one U-Net, two DeepMedics and fully convolu-

tional networks (FCNs) beat competing methods by a margin [13] and only 8 out of 57 competing methods did not apply any form of deep learning. A gradual decline of the once popular decision and random forest based methods starting in 2012 until now can be observed [13, 34, 84–86]. Especially in the last two editions, a trend of incorporating 2.5 or 3 dimensional information can be witnessed. Popular and successful attributes of methods competing in BraTS are fully convolutional architectures, architectures containing dilated convolutions and architectures containing a contracting and expanding path including skip connections. Architectures which use ensembles of the previously mentioned methods were especially successful.

For MS, there have been two popular, recent segmentation challenges. In the longitudinal lesion segmentation challenge at the International Symposium on Biomedical Imaging (ISBI) 2015, the teams on first and third place based on the mean Dice score used random forests together with an MRF, while the team on second place applied a CNN. The longitudinal lesion segmentation challenge keeps track of new submissions on a leaderboard¹. As of the 10th of April 2018, the top performing methods are all based on deep learning, including multi-dimensional gated recurrent units [6, 7] (first and fifth places), cascaded CNNs [120] (second and third place) and a multi-view CNN [15] (fifth place).

In the cross-sectional lesion segmentation challenge at the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) 2016, the top 4 ranks in terms of Dice coefficient (computed on pixels as well as whole lesions) consist of two deep learning methods, a random forest application and an unsupervised approach modelling the lesions as outliers together with a rules based approach. Methods have to be run on the organizers platform on CPU, which might have reduced the number of approaches which heavily rely on GPUs.

The 10 top performing methods in the recently held WMH segmentation challenge in conjunction with MICCAI 2017 were all deep learning based as well.

¹The leaderboard is accessible at <https://smart-stats-tools.org/node/26>.

Chapter 4

Artificial Neural Networks

Machine learning, in its most general terms, is the technique of fitting a model G with parameters θ which tries to approximate an unknown function F , mapping data x from an input domain \mathbb{I} to a target domain \mathbb{T} :

$$G_{\theta}(x) : \mathbb{I} \mapsto \mathbb{T}.$$

We are interested in the optimal setting of parameters θ of model G . In the context of supervised learning, we are given the supposed outcome $y \in \mathbb{T}$ for each input $x \in \mathbb{I}$. Using this information, we want to find a configuration for θ such that the output \hat{y} of G comes as close as possible to y . The performance can be measured using a metric M of choice, and using M as a loss function we can optimize θ such that M is minimal:

$$\min_{\theta} M(G_{\theta}(x), y),$$

where $y = F(x)$.

The classic paradigm for supervised learning tasks has been similar across different methods for a long time. One would, for given data, select meaningful features, choose a model and a classifier and train it on these features [33]. The features were constructed by hand, given some insight to the model. In more recent research, features do not need to be hand-crafted anymore but can be derived by the model based on the data itself. A popular approach for this setting is the artificial neural network (NN). In the following, we will delve into machine learning with NNs in Section 4.1. We will outline recurrent neural networks (RNNs) in Section 4.2, as they are the foundation of our method which we detail in Section 5.

4.1 Neural Networks

History of Artificial Neural Networks

In the following, we quickly summarize important developments leading to the current state of artificial neural networks. We will introduce some of the concepts which

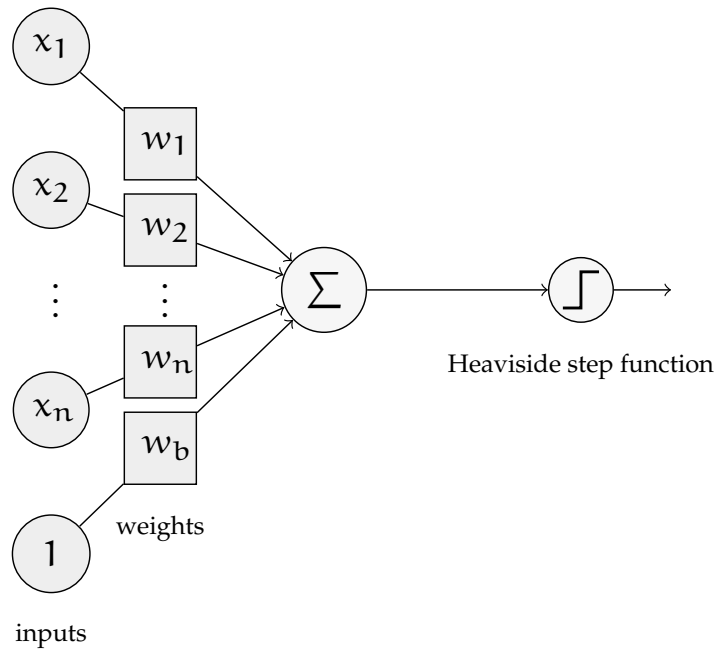


Figure 4.1: Perceptron architecture.

appear in this summary in the following sections. The first attempt to model the network of neurons present in nervous tissue was given by McCulloch and Pitts [80] in 1943, who simplified the problem and described a logical apparatus. They already distinguish between recurrent and non-recurrent, so-called feed-forward networks, where they call the recurrent networks networks with circles. They did not provide a learning algorithm, but stated that any recurrent network can be formulated as feed-forward network. Donald Hebb proposed in 1949, that often used connections of neurons are being reinforced, a fundamental operation that enables learning [50]. In 1959, Hubel and Wiesel found that the visual primary cortex consists of a cascade of simple and complex cells. Simple cells detect edges in the image, whereas complex cells also detect edges, but with a degree of spatial invariance [56].

With the Perceptron in 1957, a network without hidden layers, which was able to learn was proposed by Rosenblatt [98–100]. A perceptron could perform binary classification, thresholding a weighted sum of input numbers plus bias, as shown in Fig. 4.1. Unfortunately, by assuming that the perceptron could be used for anything, a proof of its limitations, such as the inability of modelling the XOR function [89], started what some call the *first AI winter* in 1969 [30], when research building on the perceptron almost came to a halt. With the neocognitron in 1980, a network combining local features and hierarchically stacking layers introduced the gradual integration of local features that is also used in convolutional neural networks today. Finally, in 1985 and 1986, the idea of neural networks was resurrected with the application of the backpropagation

algorithm, making learning in “multilayer perceptrons” possible [73, 101].

Research on handwritten digit recognition between 1989 and 1998 led to the first convolutional neural network with a structure still similar to today’s architectures, the lenet-5 [74]. Building on the principle of the neocognitron, it combines convolution, pooling and fully connected layers with the backpropagation algorithm to automatically classify digits. Yet problems with scaling to larger problems and model sizes and the competition introduced with the support vector machine [18] lowered interest in the approach. It wasn’t until 2006, that unsupervised pretraining was introduced [51], which made large and especially *deep* networks possible, hence the rebranding to *deep learning*, where deep refers to the number of layers that are used.

Computing on GPUs was introduced in 2009 [95] and allowed for a large speedup to conventional CPU training. Using GPUs and a sufficiently large training set, even if produced through sophisticated data augmentation, renders unsupervised pre-training unnecessary [27]. A further breakthrough for computing on the GPU and for using neural networks in general was Alexnet from Krizhevsky et al. [70], which almost halved the top 5 error rate compared to competing methods on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, an image classification benchmark proposed in 2009 [31] and commonly referred to as simply ImageNet. In addition to using two GPUs, Krizhevsky et al. applied a number of smart tricks. They started using rectified linear units instead of the classic hyperbolic tangent which partly alleviates the vanishing gradient problem and makes computation faster. Furthermore, local response normalization and dropout prevent overfitting. The large ImageNet dataset and data augmentation provide enough data to learn from. Such “tricks” seem to dominate the most recent advancements on the component level of networks, without changing the overall structure of a neural network as we know it.

ImageNet remained one of the most important benchmarks in the following years. The winners of the 2013 competition used a similar structure to Alexnet, but reduced the initial filter kernel and stride size to retain more spatial information in the network. Furthermore, they introduce a tool called “deconvnet”, which allowed to inspect which areas of an image activated which feature map [130]. A still popular pre-trained network to harvest features from is the VGG network family [110] from 2014. The network consists of only small 3x3 convolutions and max pooling operations, but contains a very large amount of parameters. The following winners of ILSVRC, the GoogLeNet or Inception architecture [116] in 2014 and the ResNet [49] in 2015 reduced the error rates to 6.67% and 3.57%, respectively. The GoogLeNet consists of so-called inception modules, which calculate a different number of differently sized convolutions and a max pooling operation and concatenate the resulting feature maps. ResNet also consists of repetitive modules, which introduce the notion of residual learning through the addition of a skip connection or identity mapping from the input of the module to the output of the module. This effectively means for the module, that it has to learn the difference to add to the input, instead of directly estimating an output. These residual connections reduce the problem of vanishing gradients, as there is always a path to skip the module as well. As a consequence, these networks consisted of hundreds of layers, depths that

were previously not possible to train.

Apart from networks for the classification task, a lot of interesting concepts and applications have been invented. For instance in the area of generative models, both the variational autoencoder (VAE) [68] and the generative adversarial network (GAN) [43] were proposed in 2014 and opened their own respective subfields of research. The VAE allows to create meaningful low-dimensional representations by encoding a distribution instead of a high dimensional value as is done in a regular autoencoder. On one hand samples from this distribution need to contain the necessary information such that a successful reconstruction of the input can be guaranteed. On the other hand, this low-dimensional distribution is constrained to be as similar as possible to a given prior distribution. These two constraints ensure that only the necessary information is encoded. The GAN is a combination of two opposing networks, the generator and the discriminator. The discriminator is given the task to learn the distribution of the training data while the generator learns to produce realistic fake imitations which resemble the training data to continuously fool the discriminator. For time series data, already in 1997 the long short-term memory (LSTM) [54] was introduced with substantial improvements in 2000 [40]. In 2014, a radical simplification of the gating structure of the LSTM termed gated recurrent unit (GRU) [23] was introduced. Both the LSTM and the GRU are being used for various tasks involving sequential data.

In the following, we will introduce a selection of the most relevant theory for neural networks in the context of this thesis.

General structure

Most neural networks can be described as a so-called feed-forward neural network (FNN). A FNN is any network whose directed computation graph does not contain loops or recurrent connections. Such a network is a combination of various small components, whose properties are well understood. Any differentiable function can theoretically be used as a component of a network.

Classical networks can be roughly split into individual *layers*. Such layers can be arbitrary, as long as the forward computation and the differentiation with respect to their input and parameters are known, but usually follow a similar structure, where optional components can just be replaced by the identity function if not needed.

Consider a network of L layers. The first component of each layer l is a mapping function $\Psi(\cdot)$, which linearly combines the inputs x_l , which consist usually at least of the outputs h_{l-1} of layer $l - 1$ or, for the first layer, the given input data:

$$z_l = \Psi_l(x_l, \theta_{\Psi_l}).$$

The weights θ_{\cdot} of these linear combinations are parameters of the network. Additionally, these mapping functions could also include further inputs, such as older intermediary outputs through skip connections or combine multiple individual strands of computation. After this linear combination, an optional normalization component $N(\cdot)$

can be applied to help optimization by reducing the *internal covariate shift* [57]:

$$\hat{z}_l = N_l(z_l, \theta_{N_l}).$$

Then, a so-called *activation function* Φ (also called *nonlinearity* or *squashing function*) is applied:

$$a_l = \Phi_l(\hat{z}_l, \theta_{\Phi_l}).$$

Without this function, all linear combinations could be expressed as one linear combination, which greatly reduces the possible functions that can be approximated.

As often used in classification networks, we can also use optional downsampling operations Γ_l :

$$h_l = \Gamma_l(a_l, \theta_{\Gamma_l}).$$

Depending on their implementation, their meaning can span from nonlinearities in max pooling operations to mapping functions with average pooling or convolution operations with a stride larger than one. The opposite, upsampling operations Γ_l^{-1} , as used for instance in autoencoders [52, 68] when applied to images and many popular semantic segmentation networks [77, 97] are usually implemented using the transposed operation of their respective downsampling counterpart.

In summary, a full layer could then be described as the application of the combined function $\Lambda_l = \Gamma_l \circ \Phi_l \circ N_l \circ \Psi_l$ to input x_l using weights $\theta_l = \theta_{\Gamma_l} \cup \theta_{\Phi_l} \cup \theta_{N_l} \cup \theta_{\Psi_l}$:

$$h_l = \Lambda_l(h_{l-1}, \theta_l) = \Gamma_l(\Phi_l(N_l(\Psi_l(x_l, \theta_{\Psi_l}), \theta_{N_l}), \theta_{\Phi_l}), \theta_{\Gamma_l}),$$

By defining x_l for each layer, for example by setting $x_l = h_{l-1}$, a network can be defined. After the last layer, we finally add a *loss function* (also *cost* or *objective function*), which defines our main objective we want to optimize our network for. Given, we are applying supervised learning and are provided with labels y , we could define a loss $\mathcal{L}(h_L, y)$. We now want to find the weights $\hat{\theta}$ that minimize this loss:

$$\hat{\theta} = \arg \min_{\theta} \mathcal{L}(h_L, y).$$

Since the concept of layers is not well defined and is hence only partly useful when defining the graph structure of an arbitrary network, we will continue our explanations with respect to the individual components making up above mentioned layers. These can be individually arranged to form any computational graph, still allowing for an end-to-end training procedure. In the following, we will use *component*, *layer*, and *node* interchangeably for one atomic element or self-contained block with a well-defined function as well as gradients with respect to both inputs and parameters.

Considering this, the whole network is differentiable using the chain rule on the individual elements. This allows us to calculate a gradient for each of the parameters in θ and use optimization techniques such as stochastic gradient descent on the network, which is commonly referred to as the *backpropagation algorithm*. For any given component $F_{\theta}(x) = h$, assuming we have a gradient $\frac{\partial \mathcal{L}}{\partial h}$ computed from some loss function \mathcal{L}

with respect to h , it is sufficient to know the gradient of F with respect to its parameters θ and its input x . Using those gradients, we can calculate the gradient of \mathcal{L} with respect to both parameters θ and input x :

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial h} \frac{\partial h}{\partial \theta}, \quad \frac{\partial \mathcal{L}}{\partial x} = \frac{\partial \mathcal{L}}{\partial h} \frac{\partial h}{\partial x}.$$

Using this recursively, we can compute the gradients of \mathcal{L} with respect to all parameters of the network (see Section 4.1).

In the following, a short introduction to each of the previously discussed components is given. In *Data*, we discuss the format and preparation of the data. We go through the mapping functions including downsampling and upsampling operations in *Mapping Functions*. We introduce a small fraction of typical activation functions in *Activation Functions*. In *Optimization* we discuss proper parameter initialization, popular choices for loss functions and detail the optimization procedure. *Regularization* discusses normalization components as well as other regularization techniques which facilitate training.

Data

The input to a network can have any shape, as long as the data can be vectorized. For instance for categorical data, we can use the so-called one-hot encoding, where we use a vector of length n , where n is the number of categories. For a given category index c , the vector takes the following form:

$$v_c(i) = \begin{cases} 1 & \text{if } i = c \\ 0 & \text{else.} \end{cases}$$

The binary case can be coded using just a scalar which is 0 if $c = 0$ and 1 if $c = 1$. Attributes, where zero to all possible values can be possible can be mapped as individual binary categorical cases.

For semantic segmentation or pixelwise classification, we usually apply the one-hot encoding described above for each pixel or voxel in the reference segmentation. For the continuous image data, we can use the training data X directly as such and are usually presented with data in the form

$$X \in \mathbb{R}^{N \times n_{0,1} \times \dots \times n_{0,d} \times c_0}$$

Since memory constraints force us to not compute on the whole data at once (and there can be advantages of not doing so), we choose *mini-batches* of B samples drawn from the full set of N samples to feed as input x_0 into the network at once. Depending on the data, we need a different number of additional (spatial) dimensions, ranging from just scalar data ($x_0 \in \mathbb{R}^{B \times c_0}$) to multiple dimensions as for instance for images ($x_0 \in \mathbb{R}^{B \times n_{0,1} \times \dots \times n_{0,2} \times c_0}$). Each $n_{l,i}$ stands for the spatial dimension i at layer l . The last dimension c_0 denotes the number of input *feature maps* or *channels*.

A number of preprocessing steps are usually necessary. First, the data is normalized, either following a distribution such as $\mathcal{N}(0, 1)$ or being in a predefined range, usually in $[-1, 1]$. For data with relative values, such as images from MRI, normalizing to zero mean and standard deviation of 1 is a good choice in our experience. If the values are absolute, squashing the values to $[-1, 1]$ using a global, sample-specific or user defined minimum and maximum can be adequate. This is for instance useful for natural images, where RGB values are in the range $0, \dots, 255$ or for Hounsfield units, which are tissue specific properties in computed tomography. More specialized normalization schemes might be necessary, depending on the type of data. For MR images, this could include bias field correction, coregistration of the data, skull stripping and more, as discussed in *Preprocessing* in Chapter 3. Especially for volumetric data or large images, where even for $B = 1$, not the whole data fits into memory, subvolumes or patches with dimensions $w_{0,1} \times \dots \times w_{0,d}$ with $w_i \leq n_i$ have to be extracted from each sample, usually at a random location. As each sample in the mini-batch, denoted as $h_{0,k}$, $k \in \{0, \dots, B-1\}$, is processed independently of the other samples, we will omit the sample index k in our notation for brevity and clarity. In each layer, its intermediary representation $h_l^{(i)}$ has c_l feature maps of (spatial) dimensions $w_{l,1} \times \dots \times w_{l,d}$.

For supervised learning, we divide our data into training, validation and test set. The training set is solely used to tune the parameters in θ . Since we rely on stochastic gradient descent and are never guaranteed, that updating our parameters results in a network state with an overall lower loss value, we use a validation set to select the best performing network parameter setting. By using the test data exclusively for the final evaluation, it is guaranteed that no information from the test data leaked into our model, even if we had access to their respective ground truths.

Mapping Functions

Mapping functions are a means of connecting the output of layer $l-1$ to the input of layer l and can consist of any linear combination of the input. We define here $h_{l-1} = x_l$ as input to layer l . We denote the intermediate result of $\Psi(h_{l-1}, W)$ as z_l . Additionally, a bias β_l can optionally be added to the mapping $\Psi(x_l, W)$, which we will omit for brevity.

Fully-Connected Layer Fully connected layers map each input vector to each output vector for each sample in the mini-batch:

$$\Psi_{FC}(x_l, W_l) = x_l W_l = z_l,$$

where $x_l \in \mathbb{R}^{b \times c_l}$, $W_l \in \mathbb{R}^{c_l \times c_{l+1}}$, $z_l \in \mathbb{R}^{b \times c_{l+1}}$ and c_i and b denote the number of neurons/channels of layer i and the mini-batch size. The gradient of the output with respect to the input and parameters is as follows:

$$\frac{\partial z_l}{\partial W_l} = x_l^T, \quad \frac{\partial z_l}{\partial x_l} = W_l^T.$$

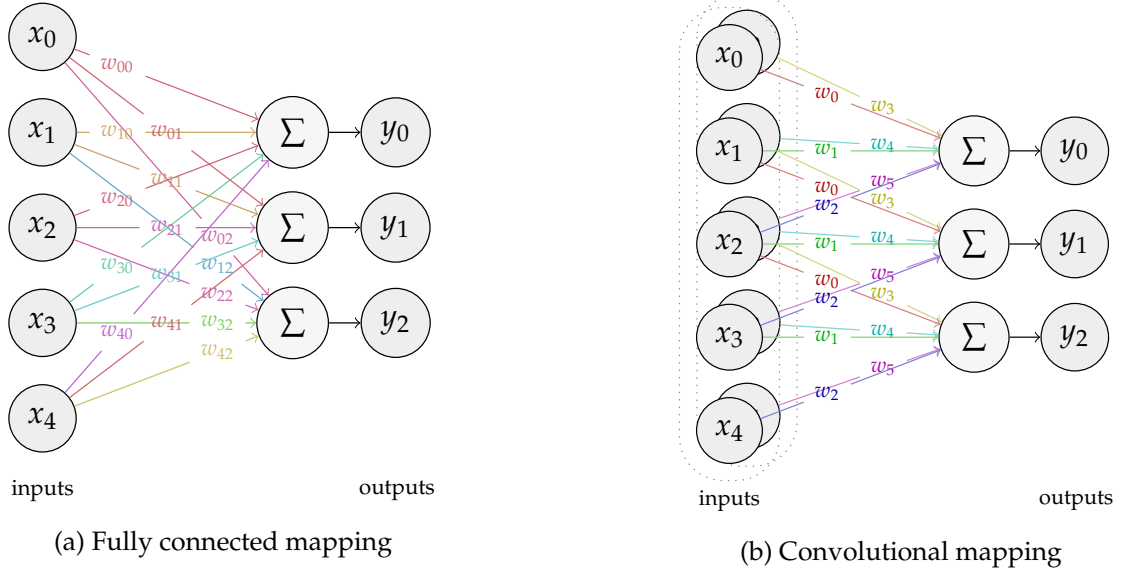


Figure 4.2: Popular mapping functions. *Left*: fully connected mapping. *Right*: convolutional mapping for one-dimensional data from two input channels (dotted grey line) onto each output channel, using a convolution kernel of 3. For clarity, we only visualize one output channel here. For multiple output channels, each output channel can be expressed in the above form with its respective, independent weights.

Fully-connected layers define a weight for every single possible connection, hence creating a lot of parameters, as shown in Fig. 4.2a. For structured data such as images, such a function can quickly lead to overfitting, as each pixel needs to define its relation with all of the other pixels independently and many more parameters are necessary compared to a *convolutional layer*.

Convolutional Layer Convolutional layers map a defined local neighborhood per location in the input to the output, with the same set of weights applied to each neighborhood in the input data. Mathematically, this corresponds to a convolution operation ($*$) (although in practice, it is usually implemented as a correlation, since for real-valued data a reversal of all convolved dimensions of one signal – e.g. the filter – is the only difference):

$$\Psi_{\text{conv}}(x_l, \omega) = \left(\sum_{i=0}^{c_l-1} x_l^{(i)} * \omega_{ij} \right)_j = z_l$$

with $x_l \in \mathbb{R}^{n_1 \times \dots \times n_d \times c_l}$, $\omega \in \mathbb{R}^{k_1 \times \dots \times k_d \times c_l \times c_{l+1}}$, $z_l \in \mathbb{R}^{n_1 \times \dots \times n_d \times c_{l+1}}$, where k denotes the filter kernel size for each dimension. The asterisk ($*$) denotes a d -dimensional convolution operation and i, j are indices for the input and output channels. An example with two input channels and one output channel is shown in Fig. 4.2b.

Pooling and Up-/Downsampling Pooling layers are used to summarize a local neighborhood to one pixel, effectively reducing the spatial resolution. A pooling layer with a pooling of $p_i = 2$ for all spatial dimensions i would hence half the spatial resolution. Popular choices for the summarization function $\gamma(\cdot)$ are the average or maximum value:

$$\begin{aligned}\Gamma_{l,p}(a_l) &= (\gamma_l((a_l)_{m_1 p_1, \dots, m_d p_d}, \dots, (a_l)_{(m_1+1)p_1-1, \dots, (m_d+1)p_d-1}))_{m_1, \dots, m_d} \\ &= h_l,\end{aligned}$$

where $a_l \in \mathbb{R}^{n_1 \times \dots \times n_d \times c_l}$, $h_l \in \mathbb{R}^{m_1 \times \dots \times m_d \times c_l}$, with $m_i = \frac{n_i}{p_i}$ for $i \in \{1, \dots, d\}$.

In e.g. semantic segmentation [77] or generative models [131], pooling destroys relevant information about the location of a feature. Pooling is hence replaced by parameterizable downsampling operations, usually in the form of a convolution layer with a stride larger than 1. A classical convolution implies a stride of one, the filter is hence applied at each pixel or voxel location with a distance of one between filter applications. Using larger strides $s > 1$ hence reduces the spatial resolution of the output by the same factor, since distances between filter applications are now of size s . This has the advantage, that an average pooling operation can still theoretically be learned, but a variable context can be defined as well as the relative importance of each pixel contributing to the neighborhood. The transpose of this operation can in turn be used to perform parameterized upsampling.

Another related concept is the atrous or dilated convolution [55, 75, 77, 128], which, with proper padding, does not reduce the spatial resolution of the output, but can take neighborhood information at variable scales into consideration and replaces e.g. shift and stitch applications for pixelwise classification. This is achieved by constructing the convolution kernel as a sparse grid of evenly spaced convolution weights, where the weights in between are set to zero.

Activation Functions

Activation functions map the input data range to a new range using a piecewise differentiable nonlinear function. Desired properties of such a function are an efficient computation of forward and backward pass, monotonicity [125] of the function and approximation of the identity near the origin [41, 48]. With some exceptions, activation functions are element-wise functions. A small selection of popular activation functions is described in the following with their respective gradient with respect to their real-valued input x .

Sigmoid The logistic or Sigmoid function is defined as

$$\begin{aligned}\sigma(x) &= \frac{1}{1 + \exp(-x)}, \\ \frac{\partial \sigma(x)}{\partial x} &= \sigma(x)(1 - \sigma(x)).\end{aligned}$$

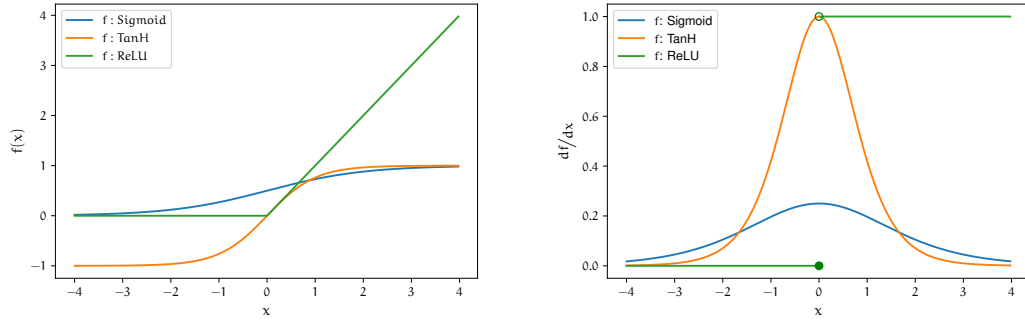


Figure 4.3: *Left*: the logistic function or Sigmoid, the hyperbolic tangent (TanH) and the rectified linear unit (ReLU). *Right*: their respective derivatives.

It maps the range of all input values to the range $(0, 1)$. As with most activation functions, the Sigmoid inherits the problem of saturation at large absolute input values, where the resulting gradient is very close to zero. Furthermore, the function takes the value 0.5 at the origin, which complicates proper weight initialization (see Section 4.1), thus making it less useful for intermediate layers.

TanH Similarly, the hyperbolic tangent (TanH) function maps the full input range to $(-1, 1)$:

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)},$$

$$\frac{\partial \tanh(x)}{\partial x} = 1 - \tanh^2(x).$$

Its advantages include the bounded values and an approximate identity mapping close to the origin, but it also suffers from saturation effects for large absolute input values.

ReLU The rectified linear unit (ReLU) [45] performs an identity mapping for positive values and maps all other values to zero:

$$\text{ReLU}(x) = \max(0, x),$$

$$\frac{\partial \text{ReLU}(x)}{\partial x} = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0. \end{cases}$$

The advantage of this activation function is its simple formula and gradient calculation, as can be seen in Fig. 4.3. However, for any input below zero, the gradient remains zero and no learning can take place. A number of activation functions try to solve this problem by approximating it using functions that have a gradient larger than zero

everywhere, such as with a continuous approximation called SoftPlus [41] or with a fixed or parametric slope for the negative part called Leaky ReLU [78] and parametric ReLU [48]. Other functions also related to ReLU focus on favorable statistical properties for optimization, such as the exponential linear unit [28] or the scaled exponential linear unit [69].

Softmax In contrast to other activation functions, the Softmax function is not an element-wise operation. From a vector of arbitrary values, it produces a vector of C probabilities representing a discrete probability distribution. It is commonly used for classification at the very end of a network to produce probabilities for each possible class i of the classifier.

$$\text{Softmax}(x, i) = \frac{\exp(x^{(i)})}{\sum_{j=0}^{C-1} \exp(x^{(j)})},$$

$$\frac{\partial \text{Softmax}(x, i)}{\partial x^{(k)}} = \begin{cases} -\text{Softmax}(x, i)\text{Softmax}(x, k) & \text{if } k \neq i \\ \text{Softmax}(x, i)(1 - \text{Softmax}(x, i)) & \text{if } k = i \end{cases}.$$

If used together with the *cross entropy loss*, the calculation of the gradient above simplifies a lot, as described in paragraph *Cross Entropy* in the next section.

Optimization

With the tools described so far, we can design a complete neural network. In the remaining part, the means to adjust the parameters of the network such that a given objective is satisfied and learning can take place are introduced. First of all, parameters have to be initialized, which is discussed in the next paragraph. Then, an objective needs to be defined, which is detailed in the paragraph *Loss Function*. In the paragraph *Backpropagation*, different methods are discussed which can be used to optimize with respect to the selected objective.

Parameter Initialization

With the introduction of unsupervised pretraining [51], training deep networks was possible. Even before this discovery, it was obvious that the success or failure of training a neural network was largely depending on the initial parameters. When state-of-the-art results were possible without pretraining, a proper parameter initialization was part of the success [70]. In general, we might want to keep the variance of the data about the same at each layer of the network. For an efficient backpropagation, the same applies to the gradient. *Glorot* initialization [41] sets the weights randomly with a variance of $\frac{2}{n_{\text{in}} + n_{\text{out}}}$ for one layer, where n_{in} and n_{out} are the number of input and output units, respectively. With the assumption that a symmetric activation function is used which is close to linear at the origin, this initialization should keep the variance of both input and output in the same range. Newer architectures, however, are

using ReLU or similar activation functions, which are not symmetric. *He* initialization [48] considers this, resulting in weights with a variance of $\frac{1}{n_{in}}$. Although *Glorot* and *He* initialization have been introduced using uniformly distributed and Gaussian distributed initial weights respectively, both uniform and Gaussian distributions are used for both initializations in practice. Other methods use orthonormal initialization [104] for deep linear networks or even constrain the network to keep the weight matrices in orthogonal form [8], due to the norm preserving property of orthogonal matrices.

Loss Function

There exist a few popular choices to design the objective of the optimization using a so-called *loss* function, depending on our requested task. If we want to use a NN for classification, we usually model the output of the network as a vector representing a discrete probability distribution. If we want to regress a vector, we do not restrict the output accordingly and choose the last activation function such that the whole range of desired values can be produced. If we want to estimate attribute probabilities, we design a vector with values from 0 to 1, but without a restriction on the sum of values. Usually, the loss function of the network is chosen based on the desired output and function of the network. Popular choices for regression and classification are the ℓ^2 -norm and the cross entropy loss respectively, which are introduced below.

Cross Entropy With C classes and B samples in the mini-batch, the cross entropy loss function is defined as

$$\mathcal{L}_{CE}(h_L, y) = \sum_{b=0}^{B-1} \sum_{c=0}^{C-1} -p(y_b = c) \log(h_{L,b}^{(c)}),$$

where $p(y_b = c)$ is the probability of target class y_b being the correct class c . When binary probabilities are used, and one clear target class y_b is given, this can be simplified considerably:

$$\mathcal{L}_{CE_{binary}}(h_L, y) = \sum_{b=0}^{B-1} -\log h_{L,b}^{(t_b)} \text{ and } t_b = \arg \max_c (y_b).$$

In conjunction with the Softmax function described above by setting

$$h_{L,b}^{(i)} = \text{Softmax}(x_{L,b}, i),$$

a number of simplifications can be made:

$$\begin{aligned}\mathcal{L}_{\text{CE-SM}}(\mathbf{x}_L, \mathbf{y}) &= - \sum_{b=0}^{B-1} \sum_{c=0}^{C-1} p(y_b = c) \log \left(\frac{\exp(x_{L,b}^{(c)})}{\sum_{j=0}^{C-1} \exp(x_{L,b}^{(j)})} \right) \\ &= - \sum_{b=0}^{B-1} \sum_{c=0}^{C-1} p(y_b = c) x_{L,b}^{(c)} + \sum_b^{B-1} \log \left(\sum_{j=0}^{C-1} \exp(x_{L,b}^{(j)}) \right), \\ \frac{\partial \mathcal{L}_{\text{CE-SM}}(\mathbf{x}_L, \mathbf{y})}{\partial x_{L,b}^{(c)}} &= -p(y_b = c) + \frac{\exp(x_{L,b}^{(c)})}{\sum_{j=0}^{C-1} \exp(x_{L,b}^{(j)})} = \text{Softmax}(x_{L,b}, c) - p(y_b = c),\end{aligned}$$

In the binary classification case, it suffices to compute only the probability h_L for the positive case, since the probability for the negative case is given by $1 - h_L$. This can be achieved with a Sigmoid function computed on one feature map and results in the following loss function, where label y_b and $p(y_b = c)$ are the same:

$$\mathcal{L}_{\text{CE}}(h_L, \mathbf{y}) = \frac{1}{B} \sum_{b=0}^{B-1} (y_b \log(h_{L,b}) + (1 - y_b) \log(1 - h_{L,b})).$$

ℓ^2 -norm The ℓ^2 -norm also features a simple gradient, independent of the last activation function.

$$\begin{aligned}\mathcal{L}_{\ell^2}(h_L, \mathbf{y}) &= \frac{1}{2B} \sum_{i=1}^B (h_{L,i} - y_i)^2, \\ \frac{\partial \mathcal{L}_{\ell^2}}{\partial h_L} &= \frac{1}{B} \sum_{i=1}^B (h_{L,i} - y_i).\end{aligned}$$

It can be used for arbitrary output, and can hence be used for regression and classification. It has been shown, however, that the cross entropy loss features less severe plateaus on its loss surface than the ℓ^2 -norm [41].

Backpropagation

As already stated in previous sections, backpropagation describes the application of the chain rule in a backwards direction to calculate the partial derivative g of a given loss function \mathcal{L} with respect to all P trainable parameters θ in a network F_θ :

$$g = \nabla \mathcal{L}(F_\theta(\mathbf{x}), \mathbf{y}) = \left(\frac{\partial \mathcal{L}(F_\theta(\mathbf{x}), \mathbf{y})}{\partial \theta_0}, \dots, \frac{\partial \mathcal{L}(F_\theta(\mathbf{x}), \mathbf{y})}{\partial \theta_{P-1}} \right).$$

Although backpropagation is to date still the basic means used to optimize a NN, it does not yet define how to use the information gathered by it. In the following, we discuss a number of popular parameter update methods using this calculated gradient.

Stochastic Gradient Descent Gradient descent (GD) is a method to find a local minimum of a function. Since the gradient of a given function $f(x)$ always points at the local steepest ascent on the function surface, iteratively following the negative direction of the gradient of a function will eventually lead us to a local minimum, if one exists and an appropriate stepsize or *learning rate* λ is chosen. At each iteration t , we can calculate the next step in the approximate direction of the minimum using the following update formula:

$$x_{t+1} = x_t - \lambda \nabla f(x_t).$$

Similarly for a classification NN, one could calculate the gradient g of the loss function $\mathcal{L}(F_\theta(x), y)$ for a network $F_\theta(x)$ with parameters θ and target labels y for all training samples and update the parameters θ accordingly:

$$\begin{aligned} \Delta\theta_t &= g, \\ \theta_{t+1} &= \theta_t - \lambda \Delta\theta_t. \end{aligned}$$

The optimum found with gradient descent can only be safely assumed to be globally optimal when the function surface is convex. This is unfortunately almost never the case with NNs. Recent research suggests though, that most local minima represent a comparably low function value [24].

The amount of data needed to train a NN does in most cases not fit entirely into memory. We are hence forced to compute only an approximation of our gradient. Using only one training sample in gradient descent is called stochastic gradient descent (SGD). In practice, we normally use a small subset of B samples from the training data, a so-called mini-batch, to calculate the gradient g_t for each iteration t during training:

$$g_t = \frac{1}{B} \sum_{b=0}^{B-1} \frac{\partial \mathcal{L}(F_{\theta_t}(x_b), y)}{\partial \theta}; \quad x_0, \dots, x_{B-1} \in \mathcal{X}.$$

This still allows to train large networks, but also enables a better approximation of the gradient compared to SGD.

Momentum Even though GD is guaranteed to converge on convex functions, it can take very long to do so. For instance at certain locations on the optimization surface, such as valleys or ridges, instead of taking the direction of steepest descent down the valley it is sometimes smarter to go along the valley. This is where *momentum* will find the best optimization direction. Momentum is the basic idea of using information from the gradient's first moment. The k -th (raw) moment of random variable R is defined as

$$\mu_k(R) = E[R^k].$$

The first moment of the gradient is hence pointing at the average direction of steepest ascent. SGD can be improved by including this first moment information. Instead

of applying the gradient updates directly, a running average over the past gradient updates is used, where γ and η scale the new gradient and the previous moment of the gradient respectively:

$$\begin{aligned}\hat{\mu}_{1,\text{mom},t}(\Delta\theta) &= \gamma g_t + \eta \hat{\mu}_{1,\text{mom},t-1}(\Delta\theta), \\ \Delta\theta_t &= \hat{\mu}_{1,\text{mom},t}(\Delta\theta).\end{aligned}$$

This is generally referred to as “using the momentum method” [101]. In the following, we will discuss different optimization techniques which will use moments extensively. We will use the following approximation for the k -th moments of R by setting $\eta = \rho$ and $\gamma = (1 - \rho)$:

$$\hat{\mu}_{k,t}(R, \rho) = \rho \hat{\mu}_{k,t-1}(R, \rho) + (1 - \rho) R_t^k,$$

where $\rho \in [0, 1]$ is a parameter usually set close to 1 and R_t is the realization of random variable R at time t .

RMSProp RMSProp is a gradient regularization method first introduced in a presentation by Tieleman and Hinton [118]. It postulates, that a per weight running average α of the squared gradient – the second moment – can help make learning much faster, as it contains information about the magnitude of each direction. Each iteration, the gradient update is hence divided by the square root of the second moment of the gradient:

$$\Delta\theta_t = \frac{g_t}{\sqrt{\hat{\mu}_2(g_t, \rho) + \epsilon}}.$$

AdaDelta Similar to RMSProp, AdaDelta [129] takes advantage of second moments. Instead of only accumulating the gradient, also the update itself is accumulated. In the original formulation, the learning rate has been factored out, such that the parameter update does not necessarily need a learning rate λ . In practice, it is still sometimes used:

$$\begin{aligned}\Delta\theta_t &= \frac{\sqrt{\hat{\mu}_2(\Delta\theta_{t-1}, \rho) + \epsilon}}{\sqrt{\hat{\mu}_2(g_t, \rho) + \epsilon}} g_t, \\ \theta_{t+1} &= \theta_t - \Delta\theta_t.\end{aligned}$$

Adam Adam [67] is a recently proposed technique which uses both first and second moment estimates of the gradient to guide the movement on the optimization surface. It is defined as follows:

$$\Delta\theta_t = \frac{\frac{\hat{\mu}_1(g_t, \beta_1)}{1 - \beta_1^t}}{\sqrt{\frac{\hat{\mu}_2(g_t, \beta_2)}{1 - \beta_2^t} + \epsilon}}.$$

Informally, this method is a mixture between the momentum method and RMSProp, where the direction of the gradient is controlled by the first moment and the magnitude is controlled by the second. Without the normalization terms $1/(1 - \beta_i^n)$, there would be a bias towards zero in the beginning of the optimization [67].

Regularization

During training of a neural network, we want to guarantee the generalizability of the network on new test data while at the same time ensuring an efficient training procedure with as few training steps as possible. There exist a number of different techniques to help with the former, the latter or both. In the following, we discuss a number of techniques we deem relevant.

Normalization Layer

Batch normalization [57] is a popular technique to improve on the speed of convergence. It is based on the idea, that learning is hindered when the distribution of the input at each layer in the network changes due to parameter updates of the previous layers during training. This effect is called the *internal covariate shift* [57]. At the batch normalization layer, the data is normalized to follow a standard normal distribution, using statistics for mean μ and standard deviation σ gathered from the mini-batch. At the same time, a running average for each mean and standard deviation is kept to be used instead of the mini-batch statistics during inference. Using trainable scale and shift parameters γ and β , any Gaussian distribution can be learned, depending on the requirements of each layer:

$$\text{BN}(x) = \gamma \frac{x - \mu}{\sigma} + \beta.$$

Similar ideas with normalization along different axes of the data tensors and using learned or mini-batch statistics during inference have been proposed, such as local response norm as used in Alexnet [70], layer normalization [9] or instance norm [119]. They are effectively trying to solve the same problem given different circumstances.

Residual Learning / Skip Connections

When approximating a function f , it can be easier to approximate a residual \tilde{f} instead of directly estimating the output of f . This is especially helpful in deep architectures, as it allows for the efficient propagation of information between elements with a large distance between each other in the network both in the forward as well as in the backward direction to benefit from each other and hence helps decreasing training time [32]. This can be achieved using a so-called skip or shortcut connection [16] to the input x . This effectively calculates what needs to be added to the input x to get the desired output h :

$$h = \lambda_r x + \tilde{f}(x).$$

The weights λ_r can be parameters of the network, but λ_r is often set to the identity and can be omitted. Residual learning gained a lot of popularity when Resnet was published [49].

Skip connections can also be implemented by concatenating data h_i from the earlier layer i to the output h_i of the previous layer to form the input x_{i+1} of the current layer

$l + 1$:

$$x_{l+1} = [h_l, \lambda_r h_l].$$

They are often used in architectures featuring a bottleneck to perpetuate high resolution information, such as in the U-net [97] for semantic segmentation.

Dropout / Dropconnect / Gaussian Multiplicative Noise

Dropout [53] is a technique of randomly picking parts of the network, which are then temporarily deactivated for one training iteration, usually random nodes including their input and output connections. Dropout can be thought of as sampling from a combinatorially large number of subsets of the network, where at each iteration, we pick one certain configuration. Applying this technique has multiple advantages. During inference, we can include all nodes and the output can be thought of as a result of an ensemble of many different networks, as each training sample was trained on a possibly unique subset of nodes. On the other hand, we can reduce overfitting, as each node can only contribute to the network with a certain probability. The most popular version of dropout consists of sampling $d_{B,i} \sim \text{Bernoulli}(p)$ from a Bernoulli distribution, which determines the inclusion of a given node i in the network:

$$\tilde{h}_l^{(i)} = \hat{f}(x_{l-1})^{(i)} \cdot \frac{d_{B,i}}{p},$$

where $\hat{f}(x_{l-1})$ is the component to be regularized and \tilde{h}_l is the regularized output of that component. By setting the output $h_l^{(i)}$ to zero for each i where $d_{B,i} = 0$, these nodes are effectively removed from the network. To not require rescaling of the outputs of all nodes by the factor $1/p$ during inference, we additionally divide d_B by p during training. Other variants include sampling from a Gaussian distribution ($d_G \sim \mathcal{N}(1, \sigma^2)$), which is then used as multiplicative noise on the nodes [112]:

$$\tilde{h}_{l,i} = \hat{f}(x_{l-1})_i \cdot d_{G,i}.$$

We can set $\sigma = \sqrt{\frac{1-p}{p}}$, such that the expected mean and variance of the Gaussian distribution matches the ones from $\text{Bernoulli}(p)$. Multiplicative Gaussian noise tends to produce slightly better results than Bernoulli dropout, since no node is completely omitted during training [112]. Instead of dropping entire nodes, it is also possible to apply the same mechanism to individual weights, which is referred to as Dropconnect [123], using either a Gaussian or Bernoulli distribution.

Data Augmentation

The amount of training data has a huge influence on the performance of most supervised machine learning algorithms. In the ideal case, we would either have as many training samples as we want to compute iterations during training, or even better, we would have a means to generate as many training samples as needed for our task. In

most cases and especially in voxelwise classification, training data is rare and producing training data is a time consuming and expensive step. In such cases, data augmentation is an effective means of generating a sufficient amount of training data. Data augmentation usually consists of different techniques that allow to create new training samples based on the available training samples. Techniques are usually selected depending on the task at hand, as different data allow for different augmentation techniques. Data can, for instance, be randomly mirrored, rotated, scaled, shifted, affinely transformed or modified by a deformation field. By selecting the optimal subset of these operations for a given dataset, we ensure that our generated training samples do not deviate significantly from our data distribution.

Which data augmentation techniques are favorable highly depends on the data. In the context of brain MR images which have been preprocessed and coregistered, mildly rotating and scaling the head will still produce realistic examples. Furthermore, a moderate deformation with a coarse grid can generate realistically looking, new brain instances. Mirroring along the sagittal plane creates credible new brain samples, mirroring along the frontal or transverse planes, however, creates instances that are not part of the same distribution.

4.2 Recurrent Neural Networks

RNNs are NNs designed to process sequential data. At each point in the sequence, the same set of calculations is executed on the previous output and current input of the sequence:

$$h_t = \phi(Uh_{t-1} + Wx_t + b), \quad (4.1)$$

where $\phi(\cdot)$ is an appropriate activation function such as \tanh and U , W and b are learnable parameters of the network. Contrary to the previously defined data structure, we model our input x_t and output h_t here as vectors with dimensionality $C_i \times B$ and $C_o \times B$, where C_i and C_o are the input and output channel sizes respectively. This allows us to formulate matrix multiplications in Eq. (4.1) in a format compatible with the literature. Equation (4.1) uses the same set of weights U , W and b at each timestep t . With this tied weights constraint, the number of parameters is greatly reduced, without gravely limiting the expressiveness of the network. Unfortunately, such a network can perform badly in practice, since the problem of *vanishing gradients* frequently occurs during backpropagation. This can easily be analyzed looking just at the gradient that is passed back through the network, if a loss function $\mathcal{L}(h_T, y_T)$ is applied to the final state of the network defined in Eq. (4.1), where we use a hyperbolic tangent as

activation function in this example:

$$\frac{\partial h_t}{\partial h_{t-1}} = (1 - \tanh(Uh_{t-1} + Wx_t + b))^2 U^T, \quad (4.2)$$

$$\frac{\partial \mathcal{L}(h_T, y_T)}{\partial h_{t-1}} = \frac{\partial h_t}{\partial h_{t-1}} \frac{\partial \mathcal{L}(h_T, y_T)}{\partial h_t}. \quad (4.3)$$

The first term of Eq. (4.3) shows the recursive relationship of future timesteps on the current timestep t . If the activation at timestep t was close to $+1$ or -1 , the gradient term would be close to zero due to Eq. (4.2). This would mean that all the accumulated gradient information of future timesteps would vanish due to the first term in Eq. (4.3) and only the gradient calculated at the current timestep would have an impact on this and past timesteps. The network will hence not be able to learn long-term relationships in the data in such situations.

LSTM

Hochreiter and Schmidhuber [54] introduced already in 1997 a method called LSTM which remedies the vanishing gradient problem. They introduce a state variable and use so-called gates which control that state and the output at each iteration t . The basic formulation of Eq. (4.1) is extended in [40, 54] as follows:

$$f_t = \sigma(U_f h_{t-1} + V_f c_{t-1} + W_f x_t + b_f), \quad (4.4)$$

$$i_t = \sigma(U_i h_{t-1} + V_i c_{t-1} + W_i x_t + b_i), \quad (4.5)$$

$$o_t = \sigma(U_o h_{t-1} + V_o c_{t-1} + W_o x_t + b_o), \quad (4.6)$$

$$\tilde{c}_t = \phi(U_c h_{t-1} + W_c x_t + b_c), \quad (4.7)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \quad (4.8)$$

$$h_t = o_t \odot \psi(c_t). \quad (4.9)$$

The first three equations above define the so-called “gates”, which serve the purpose of managing what information can enter and leave the memory cell. Equations (4.7) and (4.8) show the new state proposal \tilde{c} and the internal state c . Equation (4.9) defines the output h of the RNN. At each timestep t , the forget gate f_t (4.4) decides how much to add from the state at $t - 1$, and the input gate i_t (4.5) decides how much to add from the proposal to form the new state c_t . Said proposal is calculated from a weighed sum of different information. We chose here to include the previous output h_{t-1} , the current input x_t and a bias which is “squashed” by the activation function ϕ . Finally, the output gate o_t (4.6) decides, what linear combination of the state each node will consist of. The original publication let a lot of room for the information to be provided for the individual gates i_t , o_t and the internal state (c_t). The here shown selection of input information is a popular selection, usually either information from the previous state c_{t-1} or the previous output h_{t-1} is provided with the input x_t and an optional bias b for the weighted sum in both gates and candidate. Also, the forget gate f_t was

introduced later [40], but is now commonly used together with the other gates when talking of LSTMs. The LSTM remains to date the architecture of choice for most RNN implementations.

GRU

Recently, GRUs [23] were introduced as a simplified form of LSTMs. The GRU does not keep a separate state from the output h (4.13), making the separate state c and the output gate o in the LSTM obsolete. It combines the forget and input gates from the LSTM to one update gate z (4.11), but introduces an additional reset gate r (4.10), making it possible to omit previous state information in the new proposal \tilde{h} (4.12). It is computationally more efficient than the LSTM, as it uses less gating structure.

$$r_t = \sigma(W_r x_t + U_r h_{t-1}), \quad (4.10)$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1}), \quad (4.11)$$

$$\tilde{h}_t = \phi(W x_t + U(r_t \odot h_{t-1})), \quad (4.12)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t. \quad (4.13)$$

On selected tasks, it has been shown to perform comparable to the LSTM using the same number of parameters [25, 60]. In our own investigations in Section 5, where we adapt the GRU to the task of volumetric segmentation, we were able to show that a network with the same number of channels per multi-dimensional RNN produces similar results using either an LSTM or a GRU, but that using a GRU results in fewer parameters and faster training compared to using an LSTM [7].

Chapter 5

Multi-Dimensional Gated Recurrent Units

Introduction

We developed a general purpose supervised semantic segmentation method based on recurrent neural networks. Our contribution was the generalization of the gated recurrent unit (GRU) [23], which has been introduced for one-dimensional time signals, to data of an arbitrary number of dimensions. We proved its applicability on a volumetric brain segmentation benchmark, the MrBrains13 challenge [83], reaching 3rd place out of 37 at the time of submission. We compared ourselves to a similar work adjusting LSTM to multi-dimensional data and showed comparable performance in terms of Dice coefficient while consuming less memory and computation time. In Appendix 5.A of this chapter, we provide the formulation of MD-GRU and derive the backpropagation.

Publication We presented our approach at the 2nd workshop on *Deep Learning in Medical Image Analysis* (DLMIA), which was held with MICCAI in 2016. The submission was published in the book *Deep Learning and Data Labeling for Medical Applications*¹.

¹The publication is accessible at https://doi.org/10.1007/978-3-319-46976-8_15

Multi-dimensional Gated Recurrent Units for the Segmentation of Biomedical 3D-Data

Simon Andermatt^(✉), Simon Pezold, and Philippe Cattin

Department of Biomedical Engineering, University of Basel, Allschwil, Switzerland
`simon.anderstatt@unibas.ch`

Abstract. We present a supervised deep learning method to automatically segment 3D volumes of biomedical image data. The presented method takes advantage of a neural network with the main layers consisting of multi-dimensional gated recurrent units. We apply an on-the-fly data augmentation technique which allows for accurate estimations without the need for either a huge amount of training data or advanced data pre- or postprocessing. We show that our method performs amongst the leading techniques on a popular brain segmentation challenge dataset in terms of speed, accuracy and memory efficiency. We describe in detail advantages over a similar method which uses the well-established long short-term memory.

Keywords: Deep learning · GRU · Multi-dimensional RNN · Segmentation

1 Introduction

With the rapid advancements of imaging technologies, their ubiquitous availability and dropping prices, vast amounts of data are collected. This is particularly true for medical imaging. Accurate segmentation and delineation of e.g. pathologies in this medical data, however, pose real challenges as this is still mainly a manual process. In late phase drug studies with thousands of patients, multiple 3d datasets with different MR sequences are often collected per patient. If quantitative analysis of the immense amount of data is required, the time that has to be spent on the data by trained experts is enormous. A successful automated segmentation technique would decrease manual work to a minimum, cutting the costs and time spent on developing new treatments.

Automatic segmentation of biomedical volumetric data is, however, a challenging problem due to its high dimensionality, imaging noise, artifacts and other factors. Recent advances in the field of deep learning, especially the enabling effect of modern GPUs along with the advent of general purpose GPU computing, led to a revival of convolutional neural networks [9]. These feed-forward networks show great promise, but need a large number of layers to solve a difficult task accurately. A recurrent neural network (RNN), in contrast, can become arbitrarily deep due to its additional temporal dimension. Each timestep computed in an RNN corresponds roughly to one layer in a feed-forward network,

© Springer International Publishing AG 2016

G. Carneiro et al. (Eds.): LABELS 2016/DLMIA 2016, LNCS 10008, pp. 142–151, 2016.

DOI: 10.1007/978-3-319-46976-8_15

with the weights in one RNN being the same for each timestep. This property allows defining substantially more complexity very elegantly without the need for a huge number of layers or parameters.

The multi-dimensional Long Short-Term Memory (MD-LSTM) proposed by Stollenga et al. [13], called *PyraMiD-LSTM*, applied these insights to the Long Short-Term Memory (LSTM) [6]. It defines two LSTMs for each spatial dimension, using said spatial dimension as temporal dimension. The first one processes the data along that dimension, the second one in the opposite direction. In order to make full use of the spatial information, not only the direct predecessor along the temporal direction is taken into account, but also its local neighborhood. This can be neatly expressed using convolutions.

A relatively new RNN called Gated Recurrent Unit (GRU) [2] grew popular in recent years and became a strong competitor for the LSTM. It can be seen as a simplified version of the LSTM, which uses an update gate instead of a forget and input gate and combines the hidden and cell state [11]. It has been shown that it performs comparably to the LSTM in the task of sequence modeling [3]. Another study suggests that GRU and LSTM report similar performance on selected tasks [5]. An empirical search among more than 10 000 RNN architectures showed that on the selected tasks, although not the best performing RNN on every task, the GRU outperformed the standard LSTM architecture [8]. A larger time dimension in an RNN can mean that larger time dependencies can be represented. The lower memory requirement of the GRU means that larger volumes can be fed into the network and larger networks can be designed for the same volume size.

For all these reasons, a modification of the GRU to be able to process volumetric data seems compelling. We propose the multi-dimensional GRU (MD-GRU), which is capable of accurate segmentation of 3d data. We hint at the theoretical memory savings compared to the MD-LSTM and show that the performance of MD-GRU is comparable if not superior. Furthermore, we show that its convergence rate, computation time and combination of fewer gates favor the MD-GRU. We apply our method on a popular brain segmentation challenge dataset, achieving a score among the top 3 best performing methods.

2 Methods

2.1 Data

We used the publicly available MrBrainS [10] challenge dataset, which was one of the datasets used to evaluate the PyraMiD-LSTM. The MrBrainS challenge data consists of 5 labeled samples and 15 testing samples, where each sample has a T1 weighted, T1 inversion recovery and a FLAIR scan. The additional high-resolution T1 scan was not used, as the labeling was performed on the low resolution data. The training data contained two different label maps, one for training and one for testing. The training map consists of classes for cortical gray matter (GM), basal ganglia, white matter (WM), WM lesions, cerebrospinal fluid (CSF), ventricles, cerebellum, brainstem and background. The testing map only

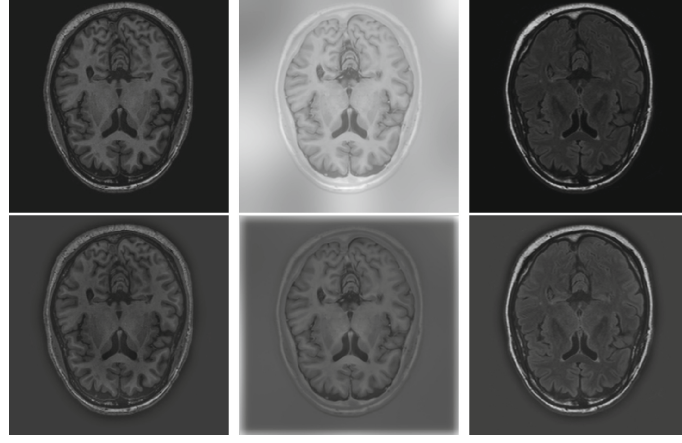


Fig. 1. Slice 19 of the 5th training sample. *Top row (left to right):* T1, T1_IR and T2_FLAIR. *Bottom row:* respective highpass filtered versions.

defines classes for GM, WM and CSF, the respective classes of the training map are merged. Brainstem and cerebellum are not included in the evaluation and do therefore not appear labeled in the testing map.

2.2 Convolutional Gated Recurrent Unit

The standard GRU as proposed in [2] is defined as

$$r^j = \sigma([W_r x]^j + [U_r h_{t-1}]^j), \quad (1)$$

$$z^j = \sigma([W_z x]^j + [U_z h_{t-1}]^j), \quad (2)$$

$$\tilde{h}_t^j = \phi([W x]^j + [U(r \odot h_{t-1})]^j), \quad (3)$$

$$h_t^j = z^j \odot h_{t-1}^j + (1 - z^j) \odot \tilde{h}_t^j, \quad (4)$$

where x is the input data, r^j is the reset gate, z^j is the update gate of the hidden unit j and the activation is performed in h^j . The operator \odot represents an elementwise multiplication. The functions $\sigma(\cdot)$ and $\phi(\cdot)$ stand for the logistic function and the hyperbolic tangent. W and U are the weight matrices for the current input and last step's output data respectively. Along the lines of Stollenga et al. [13], we adapt these equations to be able to process 3D volumes and introduce our convolutional GRU (C-GRU):

$$r^j = \sigma \left(\sum_i^I (x^i * w_r^{i,j}) + \sum_k^J (h_{t-1}^k * u_r^{k,j}) + b_r^j \right), \quad (5)$$

$$z^j = \sigma \left(\sum_i^I (x^i * w_z^{i,j}) + \sum_k^J (h_{t-1}^k * u_z^{k,j}) + b_z^j \right), \quad (6)$$

$$\tilde{h}_t^j = \phi \left(\sum_i^I (x^i * w^{i,j}) + r^j \odot \sum_k^J (h_{t-1}^k * u^{k,j}) + b^j \right), \quad (7)$$

$$h_t^j = z^j \odot h_{t-1}^j + (1 - z^j) \odot \tilde{h}_t^j, \quad (8)$$

where $*$ represents a convolution. Compared to the vanilla GRU, we introduced slight changes. We decided to use a bias b on each gate. We factored r^j out of the convolution operation between u and h_{t-1} . This change was motivated by the fact that an additional convolution would require r to have twice the support it needs now because of the chained convolution. Moreover, we reorder the data for each C-RNN such that the two spatial dimensions are closest to memory, and the temporal dimension is ordered according to the temporal direction, as explained in the next paragraph. We motivated that decision with faster possible processing speeds on the GPU, since all convolutions now require data that lies close in memory. The computations of one C-GRU are visualized as a computational graph in Fig. 2a.

The MD-GRU consists of two times D C-GRUs, where D is the dimensionality of the image data and we need one C-GRU for each of the two directions. We set the input data of channel i as $x^i \in \mathbb{R}^{S_1 \times \dots \times S_D}$. For each spatial dimension d , we create the copies $x^{i,d,-1}, x^{i,d,+1} \in \mathbb{R}^{S_d \times S_1 \times \dots \times S_D}$ of x and apply the following data transformations:

$$x^{i,d,+1}(s_d, s_1, \dots, s_D) = x^i(s_1, \dots, s_d, \dots, s_D), \quad (9)$$

$$x^{i,d,-1}(S_d - s_d, s_1, \dots, s_D) = x^i(s_1, \dots, s_d, \dots, s_D), \quad (10)$$

where s_d is the index of the assigned dimension of the C-GRU and S_d is the size of dimension d . The inverse operation is applied to $h^{j,d,+1}, h^{j,d,-1} \in \mathbb{R}^{S_d \times S_1 \times \dots \times S_D}$ to gather the final output h^j :

$$h^j(s_1, \dots, s_D) = \sum_{d=1}^D (h^{j,d,+1}(s_d, s_1, \dots, s_D) + h^{j,d,-1}(S_d - s_d, s_1, \dots, s_D)). \quad (11)$$

Figure 2b details this process for the MD-GRU. We apply the same technique for our implementation of the MD-LSTM.

2.3 Experiments

Network. We model our network similar to [13]. We include three multi-dimensional RNN (MD-RNN) layers of 16, 32 and 64 channels which are connected with pixelwise fully connected hidden layers of 25 and 45 channels respectively, each followed by a hyperbolic tangent activation function. The last MD-RNN is attached to a pixelwise fully connected layer with c channels, the same number as classes in the data. We estimate the probabilities for each class using a softmax in the last layer and consequently choose the multinomial logistic loss for the training of our network. Figure 2c shows our network setup for the case of MD-GRU.

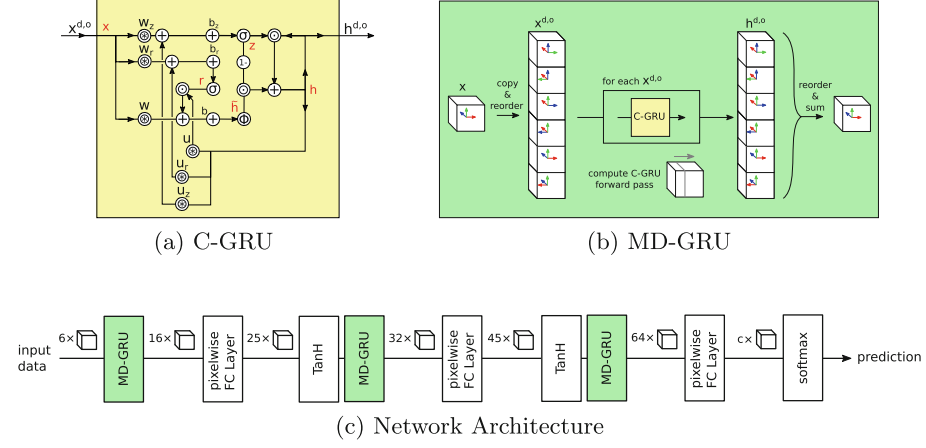


Fig. 2. (a) Directed graph denoting the computations in one C-GRU. The variables $x^{d,o}$, $h^{d,o}$ with $o \in \{-1, +1\}$ represent the input and output data across all I and J channels respectively. The \oplus operator denotes here the sum per channel j over the convolutions with each channel i or k , as used in Eqs. (5)–(7). (b) Proposed arrangement of 6 C-GRUs in a MD-GRU for three-dimensional data. (c) Setup of our network.

Setting. All experiments were calculated on an NVIDIA GTX Titan X GPU with 12 GB global memory. Our implementation of MD-LSTM and MD-GRU relied on the fast convolution routines provided by NVIDIA’s cuDNN [1]. For other layers, the already available implementations of the CAFFE¹ framework [7] were used.

Preprocessing. For all volumes, unsharp masking was done using a Gaussian smoothed image ($\sigma = 5$ voxels) which was then subtracted from the original images to produce highpass filtered volumes. The original images and the highpass filtered images were normalized to $\sigma = 1$ and $\mu = 0$, assuming normally distributed values. In this way we followed a procedure similar to [13], but omitted the histogram equalization. Figure 1 shows the original and preprocessed data for training sample 5 at slice 19.

Data Augmentation. In the training stage, at each iteration, a random location in the training data was selected and a deformation field was generated and applied to the subvolumes, which were then fed into the network. We used a procedure similar to [12], but made the grid size dependent on the data. We did not use random deformations in the feasibility study mentioned in Sect. 3.1. For the testing phase, no deformations were applied.

¹ Version 1.0.0-rc3, commit 9c46289.

Training. In three training steps we iteratively increase the subvolume size from $64 \times 64 \times 8$ voxels to $128 \times 128 \times 12$ and finally to $200 \times 200 \times 15$, keeping the third dimension smaller to account for the anisotropic MR volume resolution. We relied on AdaDelta [15] to omit the manual tuning of a learning rate. For the challenge, we additionally used DropConnect [14] of 0.5 on the input connections of each C-GRU to prevent overfitting. Training took around two days.

Testing. In the testing phase, we divided the volume into a grid of equally sized subvolumes of $120 \times 120 \times 8$, which were padded by 50, 50 and 4 voxels respectively on all sides of the volume. The padding was later used to stitch the results together using a Gaussian ($\mu = 0$, $\sigma = (10, 10, 0.8)$) to produce interpolation weights, since the borders contain starting artifacts from the individual RNNs and do not contain adequate results. Since we trained for nine classes, but only four classes were needed for the final evaluation, we simply combined the binary labels for the CSF with the ventricles, the cortical GM with the basal ganglia and the WM with the WM lesions. Everything else was considered background. Testing one volume of the MRBrainS data required 32 iterations, which needed around two minutes.

3 Results

3.1 Feasibility Study

To point out differences between the MD-GRU and the MD-LSTM, we ran the same setup with the multi-dimensional RNN layers either being an MD-GRU or an MD-LSTM. We used the first four volumes in the training set of the MrBrainS challenge and trained both networks for 3000 iterations on the largest possible resolution which was feasible for both (limited to $192 \times 192 \times 14$ by our MD-LSTM implementation). On average, one training iteration for MD-GRU and MD-LSTM took 9.1 and 12.8s, respectively. The Dice coefficients for CSF, GM, WM and ICV between the computed segmentation of the 5th training volume and the provided reference segmentation are shown in Table 1 for both the MD-GRU and MD-LSTM. Slice 19 of the computed segmentations and the reference segmentation are displayed in Fig. 3 together with a plot of a running average of 100 iterations of the loss function for each iteration of the training procedure.

Table 1. Feasibility study. Dice coefficients in percent for gray and white matter (GM/WM), cerebrospinal fluid (CSF) and intracranial volume (ICV).

	GM	WM	CSF	ICV
MD-LSTM	88.09	90.08	82.62	97.56
MD-GRU	87.88	90.15	83.19	97.73

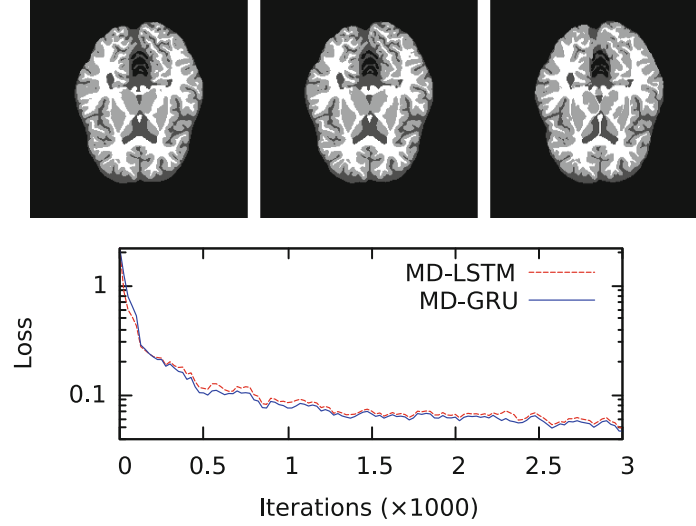


Fig. 3. Feasibility study. *Top row:* slice 19 of the 5th training volume used for the evaluation. The images from left to right represent the results of the MD-LSTM, the MD-GRU and the manual labeling. *Bottom row:* convergence rates for the feasibility study of both MD-GRU and MD-LSTM.

3.2 MD-GRU on MRBrainS

In our attempt to beat the highscore of the MRBrainS challenge, we used our described data augmentation method. Each subvolume was deformed randomly throughout all three training phases. We used all provided low resolution volumes and their highpass filtered versions. Table 2 lists our performance according to the Dice coefficients, 95th-percentile of the Hausdorff distance and average volume difference of the GM, WM, CSF and ICV. Nine measures were relevant

Table 2. MrBrainS challenge. Results of the six best performing methods for GM, WM, CSF and ICV of all three used metrics (Dice, 95th-percentile of the Hausdorff distance (HD) and average volume difference (AVD)). A bold number means best out of these six. The results reflect the state on August 12, 2016.

Team name	Rank	GM			WM			CSF			ICV		
		Dice	HD	AVD	Dice	HD	AVD	Dice	HD	AVD	Dice	HD	AVD
CU_DL2	1	86.15	1.45	6.60	89.46	1.94	6.05	84.25	2.19	7.69	98.10	2.75	1.54
CU_DL	2	86.12	1.47	6.42	89.39	1.94	5.84	83.96	2.28	7.44	97.99	3.16	1.83
MD-GRU [proposed]	3	85.40	1.55	6.09	88.98	2.02	7.69	84.13	2.17	7.44	98.15	2.37	0.86
PyraMid- LSTM2	4	84.89	1.67	6.35	88.53	2.07	5.93	83.05	2.30	7.17	98.04	2.86	0.69
FBI/LMB Freiburg [4]	5	85.44	1.58	6.60	88.86	1.95	6.47	83.47	2.22	8.63	97.98	2.51	1.06
IDSIA [13]	6	84.82	1.70	6.77	88.33	2.08	7.05	83.72	2.14	7.09	98.15	2.44	0.95

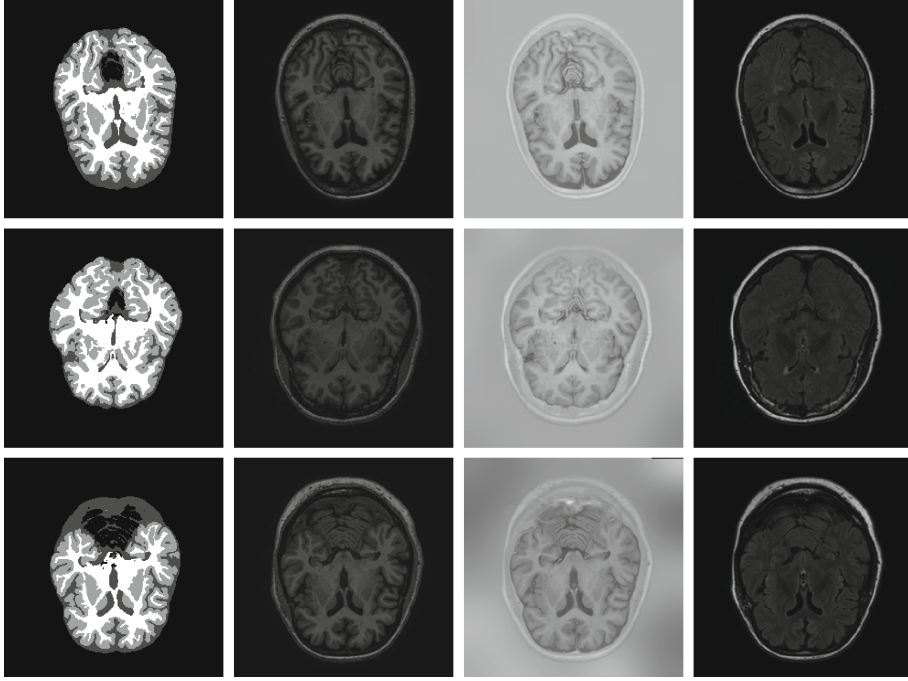


Fig. 4. MrBrainS challenge. *Rows (top to bottom):* 5th, 10th and 15th test sample. *Columns (left to right):* slice 19 of our segmentation results, T1, T1_IR and T2_FLAIR.

for the final evaluation: Dice, modified Hausdorff distance and average volume distance in each of the categories GM, WM and CSF. The sum of the ranks in these nine categories is used as the performance score and determines the final rank. Figure 4 shows the computed segmentation at slice 19 of samples 5, 10 and 15 of the test data.

4 Discussion

The feasibility study has shown that MD-GRU has great potential for the segmentation of volumetric images, since it achieved comparable results to the MD-LSTM in less time with the same settings.

Using deformation as a data augmentation strategy and DropConnect for regularization in the challenge, we ranked 3rd out of 37. Unfortunately, none of the results in the top five of the challenge highscore are published so far. The 4th and 6th entries are both incarnations of the already discussed MD-LSTM, where only the latter was described in [13] and the former likely contains unpublished improvements to their method. In contrast to [13], we did not omit the original T1_IR images. Yet some obvious misclassifications could be traced back to strong bias field artifacts in the T1_IR images. Given the small training size, using the

T1-IR images leads to apparent fitting to the bias field. Furthermore, we were not able to replicate the training volume size of Stollenga et al. [13] due to a higher memory requirement of our implementation, since we decided to copy the input and output data for each RNN layer, as detailed in Sect. 2.2. This has to be kept in mind when comparing the two approaches. Relationships between areas that are located at a certain distance in the data could therefore not be modeled in our network, where [13] was able to use the full spatial context in two dimensions as well as a larger third dimension. In their last training step more than half of the data was covered while we could only fit a bit more than a fifth in our memory.

The contribution on rank five was computed using the 3D U-Net [4]. It consists of a hierarchical convolutional neural network with shortcut connections, which is trained using various on-the-fly data augmentation techniques, including the deformation strategy used in this paper. The challenge results and corresponding adaptations of the algorithm to fit the challenge data are, however, not yet published. We believe that data augmentation is key for successful applications to problems with such a small training size.

Conclusion. With the MD-GRU, we combined the enormous expressive power of RNNs with a highly beneficial data augmentation strategy, resulting in a powerful supervised automatic segmentation technique. With a memory-savvy implementation that omits the initial reordering of the data, results surpassing the state of the art should be possible with MD-GRU.

References

1. Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., Shelhamer, E.: cuDNN: Efficient Primitives for Deep Learning. [arXiv:1410.0759](#) [cs], October 2014
2. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. [arXiv:1406.1078](#) [cs, stat], June 2014
3. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. [arXiv:1412.3555](#) [cs], December 2014
4. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. [arXiv:1606.06650](#) [cs], June 2016
5. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: A Search Space Odyssey. [arXiv:1503.04069](#) [cs], March 2015
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
7. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093* (2014)
8. Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: *Proceedings of The 32nd International Conference on Machine Learning*, pp. 2342–2350 (2015)

9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates Inc, Red Hook (2012)
10. Mendrik, A.M., Vincken, K.L., Kuijf, H.J., et al.: MRBrainS challenge: online evaluation framework for brain image segmentation in 3T MRI scans. *Comput. Intell. Neurosci.* **2015**, 16 (2015). doi:[10.1155/2015/813696](https://doi.org/10.1155/2015/813696). Article ID 813696
11. Olah, C.: Understanding LSTM Networks, August 2015. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
12. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) *MICCAI 2015. LNCS*, vol. 9351, pp. 234–241. Springer, Heidelberg (2015)
13. Stollenga, M.F., Byeon, W., Liwicki, M., Schmidhuber, J.: Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) *Advances in Neural Information Processing Systems 28*, pp. 2998–3006. Curran Associates Inc., Red Hook (2015)
14. Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R.: Regularization of neural networks using dropconnect. In: Dasgupta, S., McAllester, D. (eds.) *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, JMLR Workshop and Conference Proceedings, vol. 28, pp. 1058–1066, May 2013
15. Zeiler, M.D.: ADADELTA: An Adaptive Learning Rate Method. [arXiv:1212.5701](https://arxiv.org/abs/1212.5701) [cs], December 2012

5.A Derivation of the Forward and Backpropagation

In the following we define the internal forward and backpropagation calculation of the gates, state and output of MD-GRU, which did not fit in the original paper.

MD-GRU forward equations

We define the set D , which consists of all possible combinations of tuples $d = (a, b)$, where $a \in \{0, 1, \dots, n - 1\}$ represents the dimension of the n -d volume and $b \in \{+, -\}$ the direction. The current position in time along the specified dimension and direction is denoted by t for each C-GRU, and j denotes one channel in the feature vector. The learnable parameters $w_{d,\{z,r,\cdot\}}^j$, $u_{d,\{z,r,\cdot\}}^j$ and $b_{d,\{z,r,\cdot\}}^j$ are the weights for x , h and the bias for the update gate z , the reset gate r and the hidden state \tilde{h} of channel j in C-GRU d , respectively. The output of the MD-GRU is denoted by H . σ is the sigmoid function and ϕ denotes the hyperbolic tangent. \odot and $*$ denote the elementwise multiplication and the convolution, respectively.

$$\begin{aligned} z_{d,t} &= \sigma \left(\left(\sum_i^I (x_{d,t}^i * w_{d,z}^{i,j}) + \sum_k^J (h_{d,t-1}^k * u_{d,z}^{k,j}) + b_{d,z}^j \right)_j \right), \\ r_{d,t} &= \sigma \left(\left(\sum_i^I (x_{d,t}^i * w_{d,r}^{i,j}) + \sum_k^J (h_{d,t-1}^k * u_{d,r}^{k,j}) + b_{d,r}^j \right)_j \right), \\ \tilde{h}_{d,t} &= \phi \left(\left(\sum_i^I (x_{d,t}^i * w_d^{i,j}) + r^j \odot \sum_k^J (h_{d,t-1}^k * u_d^{k,j}) + b_d^j \right)_j \right), \\ h_{d,t} &= z_{d,t} \odot h_{d,t-1} + (1 - z_{d,t}) \odot \tilde{h}_{d,t}, \\ H &= \sum_{d \in D} h_d. \end{aligned}$$

MD-GRU backward equations

Derivation of the Gradients of the Gates, Input and Output

We assume that the gradient $\frac{\partial \mathcal{L}}{\partial H}$ of some loss function \mathcal{L} with respect to H is given. In the following, we omit \odot and assume always elementwise multiplication. We first keep in mind that the derivative of a convolution from I to J channels between some data a

and filters f with respect to a can be expressed as follows:

$$\begin{aligned} b &= \left(\sum_i^I a^i * f \right)_j, \\ \frac{\partial \mathcal{L}}{\partial a} &= \frac{\partial \mathcal{L}}{\partial b} \frac{\partial b}{\partial a} = \left(\sum_j^J \frac{\partial \mathcal{L}}{\partial b^j} * \text{flip}(f) \right)_i = \left(\sum_j^J \frac{\partial \mathcal{L}}{\partial b^j} \star f \right)_i, \end{aligned}$$

where $\text{flip}(\cdot)$ denotes reversing the direction of each dimension of the filter and \star denotes cross-correlation². We then derive the gradients of the gates:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial z_{d,t}} &= \frac{\partial \mathcal{L}}{\partial h_{d,t}} \frac{\partial h_{d,t}}{\partial z_{d,t}} = \frac{\partial \mathcal{L}}{\partial h_{d,t}} (h_{d,t-1} - \tilde{h}_{d,t}), \\ \frac{\partial \mathcal{L}}{\partial \tilde{h}_{d,t}} &= \frac{\partial \mathcal{L}}{\partial h_{d,t}} \frac{\partial h_{d,t}}{\partial \tilde{h}_{d,t}} = \frac{\partial \mathcal{L}}{\partial h_{d,t}} (1 - z_{d,t}), \\ \frac{\partial \mathcal{L}}{\partial r_{d,t}} &= \frac{\partial \mathcal{L}}{\partial \tilde{h}_{d,t}} \frac{\partial \tilde{h}_{d,t}}{\partial r_{d,t}} = \frac{\partial \mathcal{L}}{\partial \tilde{h}_{d,t}} (1 - \tilde{h}^2) \left(\sum_k^J (h_{d,t-1}^k * u_d^{k,j}) \right)_j. \end{aligned}$$

We can now use above derivatives to describe the gradients of the input:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x_{d,t}} &= \left(\sum_k^J \frac{\partial \mathcal{L}}{\partial z_{d,t}} \frac{\partial z_{d,t}^k}{\partial x_{d,t}^i} + \sum_k^J \frac{\partial \mathcal{L}}{\partial \tilde{h}_{d,t}} \frac{\partial \tilde{h}_{d,t}^k}{\partial x_{d,t}^i} \right)_i \\ &= \left(\sum_k^J \left(\left(\frac{\partial \mathcal{L}}{\partial z_{d,t}} (1 - z_{d,t}^k) z_{d,t}^k \right) * \text{flip}(w_{d,z}^{k,i}) \right) \right)_i \\ &\quad + \left(\sum_k^J \left(\left(\frac{\partial \mathcal{L}}{\partial \tilde{h}_{d,t}} (1 - (\tilde{h}_{d,t}^k)^2) \right) * \text{flip}(w_d^{k,i}) \right) \right)_i \\ &\quad + \left(\frac{\partial \mathcal{L}}{\partial r_{d,t}} (1 - r_{d,t}^k) r_{d,t}^k * \text{flip}(w_{d,r}^{k,i}) \right)_i, \end{aligned}$$

²Usually, deep learning frameworks perform the forward pass using cross-correlation, which then results in a convolution during backpropagation. In practice, this only results in flipped filter representations for real input data, as the filters are learned from scratch in both cases anyway.

and the gradients of the output of the previous timestep:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial h_{d,t-1}} &= \frac{\partial \mathcal{L}}{\partial h_{d,t}} \frac{\partial h_{d,t}}{\partial h_{d,t-1}} + \frac{\partial \mathcal{L}}{\partial H} \frac{\partial H}{\partial h_{d,t-1}} \\
&= \frac{\partial \mathcal{L}}{\partial h_{d,t}} \frac{\partial h_{d,t}}{\partial z_{d,t}} \frac{\partial z_{d,t}}{\partial h_{d,t-1}} + \frac{\partial \mathcal{L}}{\partial h_{d,t}} \frac{\partial h_{d,t}}{\partial \tilde{h}_{d,t}} \frac{\partial \tilde{h}_{d,t}}{\partial h_{d,t-1}} \\
&\quad + \frac{\partial \mathcal{L}}{\partial h_{d,t}} z_{d,t} + \frac{\partial \mathcal{L}}{\partial H_{t,d}} \cdot 1 \\
&= \left(\sum_k^J \left(\left(\frac{\partial \mathcal{L}}{\partial z_{d,t}} (1 - z_{d,t}^k) z_{d,t}^k \right) * \text{flip} \left(u_{d,z}^{k,j} \right) \right) \right)_j \\
&\quad + \left(\sum_k^J \left(\left(\frac{\partial \mathcal{L}}{\partial r_{d,t}} (1 - r_{d,t}^k) r_{d,t}^k \right) * \text{flip} \left(u_{d,r}^{k,j} \right) \right. \right. \\
&\quad \left. \left. + \left(\frac{\partial \mathcal{L}}{\partial \tilde{h}_{d,t}} r_{d,t}^k \left(1 - (\tilde{h}_{d,t}^k)^2 \right) \right) * \text{flip} \left(u_d^{k,j} \right) \right) \right)_j \\
&\quad + \frac{\partial \mathcal{L}}{\partial h_{d,t}} z_{d,t} + \frac{\partial \mathcal{L}}{\partial H_{t,d}}.
\end{aligned}$$

The above derivations of the gates and outputs can then be used alternatingly to perform the full backpropagation back to time $t = 0$.

Derivation of the Gradients of the Weights

With the individual partial derivatives of \mathcal{L} with respect to the three gates at each timepoint t , we can derive gradients for each used parameter. Since we share weights across timesteps, we take the sum of individual contributions over all timesteps:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_{d,z}^{k,i}} &= \sum_t \left(x_{d,t}^i * \left(\frac{\partial \mathcal{L}}{\partial z_{d,t}^k} z_{d,t}^k (1 - z_{d,t}^k) \right) \right), \\
\frac{\partial \mathcal{L}}{\partial w_{d,r}^{k,i}} &= \sum_t \left(x_{d,t}^i * \left(\frac{\partial \mathcal{L}}{\partial r_{d,t}^k} r_{d,t}^k (1 - r_{d,t}^k) \right) \right), \\
\frac{\partial \mathcal{L}}{\partial w_d^{k,i}} &= \sum_t \left(x_{d,t}^i * \left(\frac{\partial \mathcal{L}}{\partial \tilde{h}_{d,t}^k} \left(1 - (\tilde{h}_{d,t}^k)^2 \right) \right) \right), \\
\frac{\partial \mathcal{L}}{\partial u_{d,z}^{k,j}} &= \sum_t \left(h_{d,t-1}^j * \left(\frac{\partial \mathcal{L}}{\partial z_{d,t}^k} z_{d,t}^k (1 - z_{d,t}^k) \right) \right), \\
\frac{\partial \mathcal{L}}{\partial u_{d,r}^{k,j}} &= \sum_t \left(h_{d,t-1}^j * \left(\frac{\partial \mathcal{L}}{\partial r_{d,t}^k} r_{d,t}^k (1 - r_{d,t}^k) \right) \right), \\
\frac{\partial \mathcal{L}}{\partial u_d^{k,j}} &= \sum_t \left(h_{d,t-1}^j * \left(\frac{\partial \mathcal{L}}{\partial \tilde{h}_{d,t}^k} \left(1 - (\tilde{h}_{d,t}^k)^2 \right) r_{d,t}^k \right) \right),
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial b_{d,z}^j} &= \sum_t \left(\frac{\partial \mathcal{L}}{\partial z_{d,t}^j} z_{d,t}^j (1 - z_{d,t}^j) \right), \\
\frac{\partial \mathcal{L}}{\partial b_{d,r}^j} &= \sum_t \left(\frac{\partial \mathcal{L}}{\partial r_{d,t}^j} r_{d,t}^j (1 - r_{d,t}^j) \right), \\
\frac{\partial \mathcal{L}}{\partial b_d^j} &= \sum_t \left(\frac{\partial \mathcal{L}}{\partial \tilde{h}_{d,t}^j} \left(1 - (\tilde{h}_{d,t}^j)^2 \right) \right).
\end{aligned}$$

Chapter 6

Supervised Lesion Segmentation using MD-GRU

Introduction

We adapted our initial formulation of MD-GRU using techniques from the deep learning community and evaluated modifications to our formulation with the ultimate goal to segment MS lesions in the brain. We assessed the different modifications on the training data of the LMSLS challenge [22] from 2015, an open benchmark for lesion segmentation methods. Using the best-performing subset of modifications to our algorithm, we were able to achieve first place on the challenge and held that position for over a year, despite 10 newer entries appearing in the top 25 in that time. Derivations of this method were also applied to the WMHs challenge and the BraTS challenge held with MICCAI 2017. We were able to reach second place in the WMHs challenge, whereas the final rank on BraTS is still not published.

Publication We presented our approach at the 3rd workshop on *Brain-Lesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries* (Brainles) which was held at MICCAI in 2017. The paper was published in the book *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*¹.

¹The publication is accessible at https://doi.org/10.1007/978-3-319-75238-9_3

Automated Segmentation of Multiple Sclerosis Lesions Using Multi-dimensional Gated Recurrent Units

Simon Andermatt^(✉), Simon Pezold, and Philippe C. Cattin

Department of Biomedical Engineering, University of Basel, Allschwil, Switzerland
`simon.anderstatt@unibas.ch`

Abstract. We analyze the performance of multi-dimensional gated recurrent units on automated lesion segmentation in multiple sclerosis. The segmentation of these pathologic structures is not trivial, since location, shape and size can be arbitrary. Furthermore, the inherent class imbalance of about 1 lesion voxel to 10 000 healthy voxels further exacerbates the correct segmentation. We introduce a new MD-GRU setup, using established techniques from the deep learning community as well as our own adaptations. We evaluate these modifications by comparing them to a standard MD-GRU network. We demonstrate that using data augmentation, selective sampling, residual learning and/or DropConnect on the RNN state can produce better segmentation results. Reaching rank #1 in the ISBI 2015 longitudinal multiple sclerosis lesion segmentation challenge, we show that a setup which combines these techniques can outperform the state of the art in automated lesion segmentation.

Keywords: MD-GRU · MDGRU · Automatic MS lesion segmentation

1 Introduction

Multiple sclerosis (MS) is a frequent disease of the central nervous system, which prevalently occurs in young adults, especially in women. The evaluation of lesions in the brain is part of the clinical diagnostic procedure and is important when evaluating medical trials for new treatments. The manual segmentation of lesions, especially on high-resolution 3d scans, is very time consuming as well as prone to errors due to inter- and intra-rater variability [5]. Recently, recurrent neural networks (RNN) have shown the capability to match the state of the art in brain segmentation. In the brain segmentation benchmark used in [1, 10], three of the top six methods are based on RNN. Not only their performance, but also the elegant way of describing data with tied weights do speak for them, since fewer parameters have to be used for the model. We take a closer look at the multi-dimensional gated recurrent unit (MD-GRU) [1] due to its high ranking on the MRBrains challenge. Lesions, as any pathology, are hard to model. We hence treat lesion segmentation independently from anatomy segmentation and

consider to reevaluate some findings in [1,10] in the context of lesion segmentation. In the following, we explore different extensions to the MD-GRU with the focus on improvements on lesion segmentation. We investigate some design choices made in the original publication of the MD-GRU [1] and apply emerging deep learning techniques which proved to be effective. We investigate our adaptations on the training data of a publicly available challenge dataset. We then use the best performing combination of our modifications and apply it on the full dataset. Our implementation can be found at <https://github.com/zubata88/mdgru>.

2 Materials and Methods

2.1 Longitudinal MS Lesion Segmentation Challenge (ISBI 2015)

The longitudinal MS lesion segmentation challenge [2] was held in conjunction with ISBI 2015, but the data and challenge is still available online for further use. The data consists of 5 training patients and 15 test patients, with 4 to 6 screenings each, consisting of an MPRAGE, a T2, a PD and a FLAIR sequence. In all of our experiments, we only incorporate the provided preprocessed MR data and their high-pass filtered counterparts (see Sect. 2.3), as shown in Fig. 1. The remaining screenings for the first patient in the training data are shown in Fig. 2. For the training data, each screening of each patient holds two segmentation masks. Segmentation masks for the test data are not available, but binary predictions can be evaluated automatically on the challenge website.

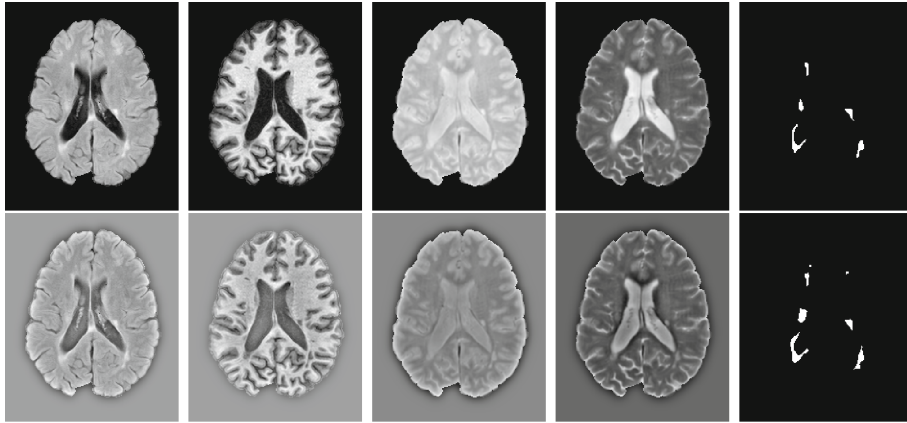


Fig. 1. Slice 90 of the baseline scan of the first training sample. *Top row (left to right):* FLAIR, MPRAGE, PD, T2 scan and label mask of rater 1. *Bottom row:* respective high-pass filtered versions and label mask of rater 2.

2.2 Original MD-GRU Setup

In the following, we define the peculiarities of MD-GRU [1] that are relevant for the evaluation of our modifications to the model. Equation (1) denotes channel j of output H , which consists of the sum of all outputs of the $N \cdot D$ individual convolutional gated recurrent units (C-GRUs) it is made of. Each C-GRU computes the data along either the forward or backward direction n of dimension d :

$$H^j(x) = \sum_{n \in \{1, -1\}} \sum_d^D h^{j,n,d}. \quad (1)$$

The following are the original C-GRU equations [1]. Index t denotes the timestep and iterates over the slices along d in direction n . Since the computations of each C-GRU are independent, we omit the indices for n, d for better readability in the following:

$$r^j = \sigma \left(\sum_i^I (x^i * w_r^{i,j}) + \sum_k^J (h_{t-1}^k * u_r^{k,j}) + b_r^j \right), \quad (2)$$

$$z^j = \sigma \left(\sum_i^I (x^i * w_z^{i,j}) + \sum_k^J (h_{t-1}^k * u_z^{k,j}) + b_z^j \right), \quad (3)$$

$$\tilde{h}_t^j = \phi \left(\sum_i^I (x^i * w^{i,j}) + r^j \odot \sum_k^J (h_{t-1}^k * u^{k,j}) + b^j \right), \quad (4)$$

$$h_t^j = z^j \odot h_{t-1}^j + (1 - z^j) \odot \tilde{h}_t^j. \quad (5)$$

The indices i and j, k denote the respective input and output channels. Variables u, w and b are trainable weights. We refer to Eqs. (2) and (3) as *reset* and *update gate*, Eq. (4) as *proposal* and Eq. (5) as *output* or *state*.

To analyze the influence of different adjustments, we will use a standard network, similar to the one published in [1]. It consists of 3 layers of MD-GRUs of 16, 32 and 64 channels, which are connected with voxelwise fully connected layers with biases consisting of 25 and 45 channels followed by a tanh activation function. The last MD-GRU is connected to a voxelwise fully connected layer of c channels, one for each class. Finally, a softmax layer is applied and the network is trained minimizing the negative log likelihood. Equation (6) summarizes the setup, where superscript numbers denote the number of channels at each layer and subscripts enumerate the independent layers of the same type:

$$h = \text{softMax}(\text{conv}_3^c(\text{H}_3(\tanh(\text{conv}_2^{64}(\text{H}_2(\tanh(\text{conv}_1^{45}(\text{H}_1(\tanh(\text{conv}_1^{25}(\text{H}_1(x))))))))))). \quad (6)$$

2.3 Evaluated Design Choices

The MD-GRU showed promising results with a relatively simple architecture. In the following we motivate and evaluate modifications to the original architecture.

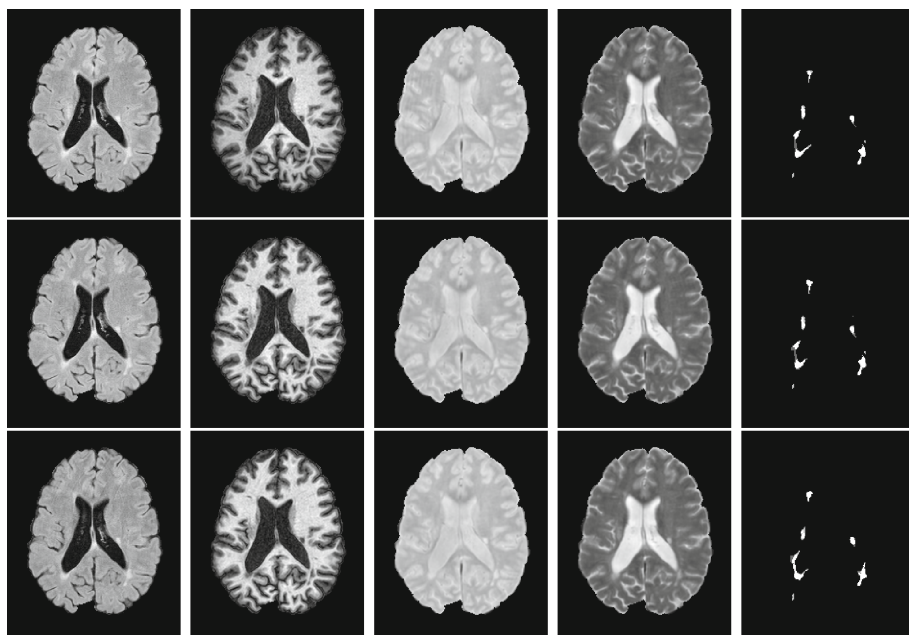


Fig. 2. Slice 90 of each followup scan of first training sample. *From left to right:* FLAIR, MPRAGE, PD, T2 scan and combined rater mask.

High-Pass Filtering. A high-pass filter was applied to the images by subtracting a Gaussian filtered version of the image volumes from the original volumes. Especially in situations with almost piecewise constant functions, such as MR images of the brain, this preprocessing step can help “announcing” a change of tissue before it actually happens, as can be seen for instance around the masked brains in Fig. 1. In Fig. 3, we inspect the voxel values along one anteroposterior line through the volume. In our experiments, we investigate, how much high-pass filtered data can help detract the influence of low frequency intensity changes in the data.

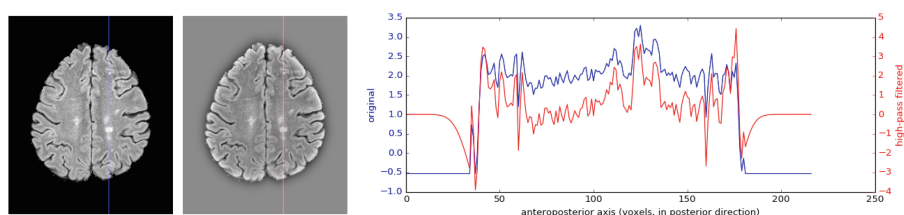


Fig. 3. Impact of high-pass filtering on the fourth screening of the sixth training sample. *Left:* Slice 110 of original and high-pass filtered FLAIR scan. *Right:* Plot of marked red and blue lines on the left for both images after normalization. (Color figure online)

Reset Gate Location. Compared to the original formulation of the GRU [3], the C-GRU applies the reset gate at a slightly different position, as depicted in Fig. 4a. In the GRU, the reset gate r is directly multiplied to the previous output h_{t-1} :

$$\tilde{h}_t^j = \phi([Wx]^j + [U(r \odot h_{t-1})]^j) \quad (7)$$

In the C-GRU however, it is multiplied to the result of the convolution of the previous output h_{t-1} with u , as shown in Eq. (4).

The provided motive for this decision is, that r is the result of convolutions and already contains information of its neighbors. This effectively means that the reset gate of channel j only directly affects the proposal of channel j instead of all proposals. We evaluate this decision by comparing to a modified C-GRU, which more closely follows the original formulation:

$$\hat{h}_t^j = \phi \left(\sum_i (x^i * w^{i,j}) + \sum_k ((r^{k,j} \odot h_{t-1}^k) * u^{k,j}) + b^j \right). \quad (8)$$

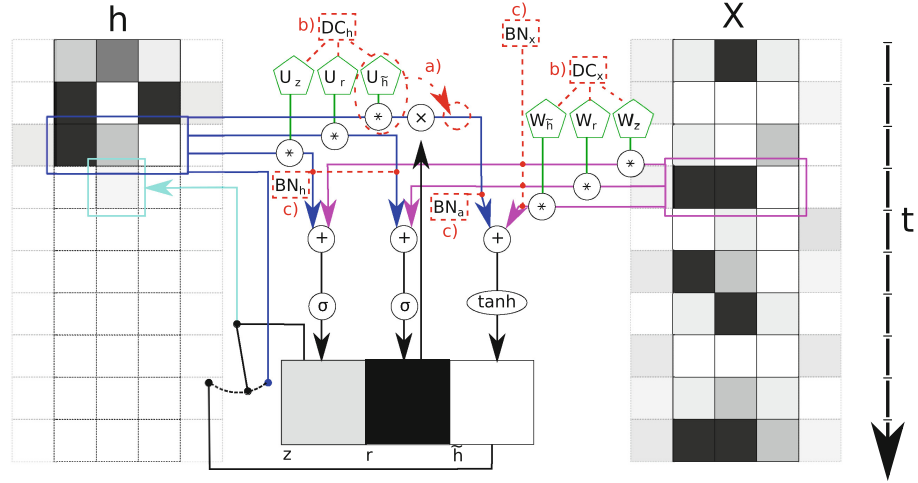


Fig. 4. Schematic and computational graph of a C-GRU with one-dimensional filters. The proposed changes are marked with dashed red lines: (a) the order of the reset gate application, (b) DropConnect at state and input weights and (c) batch normalization at input, gate states and proposal activation states. (Color figure online)

Contribution Weights for Individual C-GRU Outputs. In the original MD-GRU formulation, the individual C-GRU outputs are simply summed to gather the result H . As already implemented in the first bidirectional RNN [9], the states for each direction could be weighted independently, resulting in a

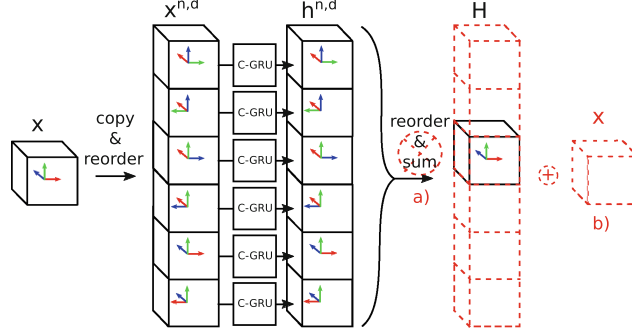


Fig. 5. Composition of an MD-GRU. *The proposed changes are marked with dashed red lines: (a) leaving out the sum of the individual directional states or (b) adding residual learning at the MD-GRU level. (Color figure online)*

more complex model. We investigate the potential benefit of concatenating the C-GRU outputs, thereby in our case of 3d volumes increasing the number of output channels sixfold. Figure 5a shows this on the example of MD-GRUs that handle volumetric data.

DropConnect. Instead of dropout, a similar method called DropConnect (DC) [11] is used at a constant rate of 0.5, which drops weights instead of outputs. In the original formulation [1], we decided to implement DC on the input weights in MD-GRU and to use a fixed drop rate of 0.5. Dropout has been reported to not work well on MD-LSTM [10] and applying it on the state in RNN has been advised against [12]. We analyze the effect of applying DC on input, state or both using different drop rates (Fig. 4b).

2.4 Techniques to Improve Accuracy and Shorten Training Time

Batch/Instance Normalization. The first technique we investigate is batch normalization (BN) [7]. BN allows for higher learning rates and faster convergence, thereby drastically reducing the training time. By normalizing the input to activations, the so-called covariate shift is reduced. This enables a layer further down the network to learn more independently from the layers before it. We build on the results on BN in one-dimensional RNN in [4] and define BN as

$$BN(x, \gamma) = \gamma \frac{x - \hat{\mu}}{\sqrt{\hat{\sigma} + \epsilon}} + \beta, \quad (9)$$

where we set β to 0 at any place, due to the biases that are already in place [4]. Due to our inherent mini-batch of one (we can only process one subvolume at a time per training iteration), we calculate the statistics for mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$ on the whole data per channel. Since the data in the subvolumes

is heavily correlated, we calculate our training statistics for each training iteration k from the m most recent training samples with $\hat{\mu}^{(k)} = \frac{1}{m} \sum_{q=k-m+1}^k \mu^{(q)}$ and $\hat{\sigma}^{(k)} = \frac{1}{m} \sum_{q=k-m+1}^k \sigma^{(q)}$. We keep a separate exponential moving average for both mean and standard deviation over all training samples to be used for testing. We apply the following BN:

$$r^j = \sigma(BN_x(\sum_i^I (x^i * w_r^{i,j}), \gamma_x) + BN_h(\sum_k^J (h_{t-1}^k * u_r^{k,j}), \gamma_h) + b_r^j), \quad (10)$$

$$z^j = \sigma(BN_x(\sum_i^I (x^i * w_z^{i,j}), \gamma_x) + BN_h(\sum_k^J (h_{t-1}^k * u_z^{k,j}), \gamma_h) + b_z^j), \quad (11)$$

$$\tilde{h}_t^j = \phi(BN_x(\sum_i^I (x^i * w^{i,j}), \gamma_x) + BN_a(r^j \odot \sum_k^J (h_{t-1}^k * u^{k,j}), \gamma_a) + b^j), \quad (12)$$

where we keep individual statistics for the input convolutions, the gate state convolutions and the state convolution of the proposal. Figure 4c shows the different locations we apply BN at.

Residual Learning. Using skip connections allowed ResNet [6] to ascend on top of a number of ILSVRC & COCO 2015 competitions, as it reportedly allows for deeper networks and faster convergence. We introduce skip connections linking input and output of each MD-GRU (Fig. 5b), allowing the network to choose between learning a residual or ignoring the previous input. We evaluate the following adjustment in between individual MD-GRU layers:

$$H_{res}(x) = \text{conv111}(x) + \overset{c}{H}(x), \quad (13)$$

where c denotes the number of output channels of H and the additional convolution increases the input channels to c . We refrain from applying additional

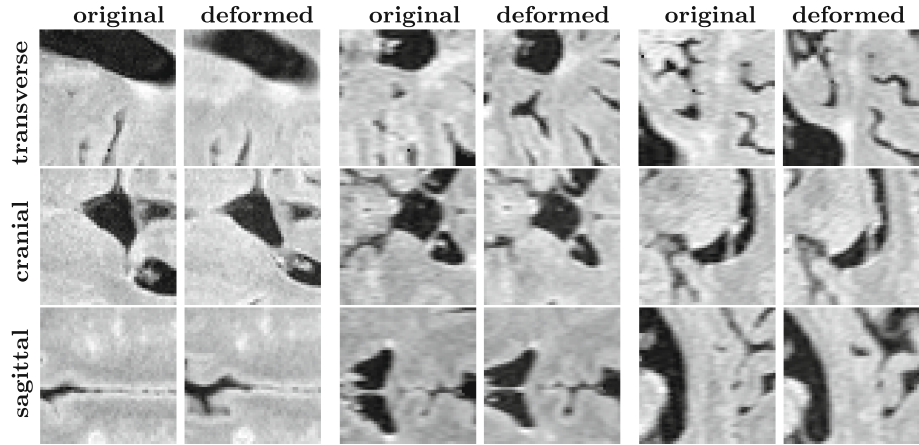


Fig. 6. Examples of random deformations performed on 48^3 subvolumes. Rows: transverse, coronal and sagittal planes of the original (left) and deformed (right) sample. Columns: individual random samples.

residual learning inside the MD-GRU, as it has been shown that residual networks with shared weights can be reformulated as plain RNN [8].

Data Augmentation. Data augmentation can have beneficial effects on networks which are trained with little data [1]. On a low resolution grid of spacing f voxels, we draw three values from a normal distribution $\mathcal{N}(0, 5)$. Using cubic interpolation, we create a smooth deformation field, which we apply on the voxelwise sampling of the subvolumes used for training (Fig. 6).

3 Experiments and Results

We train each model for 3000 iterations, which will not produce competitive results, but should give an indicator of how much an adjustment affects the method. We only included the first 4 baseline scans during training and evaluate all combinations on the baseline scan of the 5th training patient. Using only 4 samples, we remove any redundancy in the data and decrease the search space, allowing for faster convergence. For all experiments, we train on 48^3 random samples and create a full volume during the test phase by stitching patches of 32^3 with a padding of 8 together.

Data Sampling Techniques and MD-GRU Design Evaluation. We considered the following different data sampling techniques. To quantify the impact of the high-pass filtered data, we trained networks with only the original data, only the high-pass filtered data and both. We further analyzed the impact of data augmentation through random deformation, varying the size f of the deformations. We evaluate the impact of forcing every second random training sample to contain lesions (selective sampling). We also evaluate potentially not summing the individual C-GRU results and the misplacement of the reset gate as defined in Eq. (12). The respective results are listed in Table 1A.

Table 1. Summary of the different combinations which were trained on the first four and evaluated on the fifth training sample of the ISBI challenge data. Dice coefficients are provided in percent and bold face denotes results that were better than those of the baseline architecture (A, top row). The two provided masks we compare ourselves to are denoted as M1 and M2.

A	Dice		B	Dice		C	Dice			
	M1	M2		M1	M2		M1		M2	
Baseline [1]	38.18	37.70	DC(0.5,h,x)	29.39	26.35		m=16	m=1	m=16	m=1
Only original	0.00	0.00	DC(0.5,h)	44.06	41.55	BNx	31.77	30.58	26.60	24.94
Only filtered	28.30	25.65	No DC	27.85	23.54	BNx+DC(x)	37.53	20.92	32.19	18.45
Def($f = 24$)	43.74	36.94	DC(0.75,x)	21.27	17.24	BNh	31.78	29.42	30.03	31.98
Def($f = 48$)	48.43	49.12	DC(0.75,x,h)	15.79	13.63	BNh+DC(x)	28.95	30.13	26.03	27.74
			DC(0.75,h)	23.27	20.13					
Selective samples	44.95	40.70	DC(0.875,x)	21.82	19.33	BNa	3.05	3.04	2.50	2.49
No MD-GRU sum	17.79	15.39	DC(0.875,h)	19.87	17.29	BNa+DC(x)	9.70	10.20	8.59	9.02
Misplaced r	24.75	21.20	RL	41.51	35.71					

Table 2. Mean and standard deviation of the crossvalidation with the Dice coefficient in percent, the Hausdorff distance (HD) as well as the average volume distance (AVD) with best scores and lowest standard deviations in bold face.

	Dice		HD		AVD	
	M1	M2	M1	M2	M1	M2
Baseline [1]	20.03 \pm 14.13	19.86 \pm 13.17	39.82 \pm 7.62	39.15 \pm 6.69	7.34 \pm 7.89	6.79 \pm 6.52
DC(h)	33.47 \pm 8.57	32.93 \pm 8.82	35.84 \pm 5.87	36.64 \pm 4.40	5.78 \pm 6.16	5.64 \pm 5.58
Residual learning	35.95 \pm 13.78	35.10 \pm 10.19	40.61 \pm 9.45	36.29 \pm 6.65	7.27 \pm 6.90	6.61 \pm 5.48
Selective sampling	44.10 \pm 4.55	40.54 \pm 4.64	41.32 \pm 3.15	40.31 \pm 4.57	5.07 \pm 5.42	4.87 \pm 4.75
Def ($f = 48$)	38.29 \pm 21.51	34.70 \pm 19.82	37.27 \pm 8.89	38.57 \pm 8.09	2.91 \pm 3.35	2.64 \pm 2.43
All of the above	62.85 \pm 15.31	55.24 \pm 13.66	32.60 \pm 8.58	29.82 \pm 4.72	1.83 \pm 1.22	2.18 \pm 1.73

DropConnect, Batch Normalization and Residual Learning. Since both DC and BN act as regularization, we evaluated them both jointly and individually. In Table 1B, we list Dice coefficients obtained using different DC settings on input x and/or state h with the designated drop rate. At the bottom, we list the result obtained by applying residual learning (RL) in between MD-GRU layers. In Table 1C, the Dice coefficients resulting from different BNs both with and without simultaneous DC with a drop rate of 0.5 on the input are shown.

Putting it All Together. In a last experiment, we performed leave-one-out crossvalidation with the modifications which performed better than the standard network (Table 1) and the sum of those modifications. Using the Dice as performance measure, the selected techniques were random deformation with a grid spacing of 48, selective sampling, residual learning and DC on the state instead of the input. The crossvalidation results can be found in Table 2.

3.1 Improved Network

So far, we restricted ourselves to a subset of the data and only 3000 training iterations. For the final evaluation on the challenge website, we considered the complete training data, instead of just using the first scan of each patient. We decided to use a combination of data augmentation through random deformation ($f = 75$) and subvolumes of 80^3 together with DC on the state h , residual learning and selective sampling. We merged rater masks by creating 4 classes, one for each label combination during training and assumed a lesion voxel during inference, when its probability for background was below 0.5. We trained the network for 10000 iterations and managed to achieve first place in the ISBI challenge. Figure 7 shows slice 80 of the best and worst segmentation, judging from the challenge score computed on both raters. The first five entries of the challenge are listed in Table 3 with the mean of each metric that contributes to the challenge score. The fifth entry was created using the MD-GRU as described in its original

publication [1] and 40 000 training iterations. Unfortunately, none of the other competing entries have been published yet, which makes a comparison of the methods impossible.

Table 3. The five best scoring methods of the longitudinal MS lesion segmentation challenge with challenge score, volume correlation (VC), Dice coefficient, positive predictive value (PPV), lesion false positive rate (LFPR), lesion true positive rate (LTPR). Dice, PPV, LFPR and LTPR are denoted in percent and best values out of five are printed in bold. In brackets we denote the relative weight of each metric on the final score.

	Score	VC (1/4)	Dice (1/8)	PPV (1/8)	LFPR (1/4)	LTPR (1/4)
asmsl (proposed)	92.076	0.862	62.98	84.46	20.13	48.71
nic_vicorob_test	91.440	0.840	64.29	79.25	15.46	38.72
VIC_TF_FULL	91.331	0.866	63.05	78.67	15.29	36.40
MIPLAB_v3	91.267	0.823	62.74	79.97	23.17	45.40
miac_results [1]	91.011	0.867	66.78	74.05	40.73	58.29

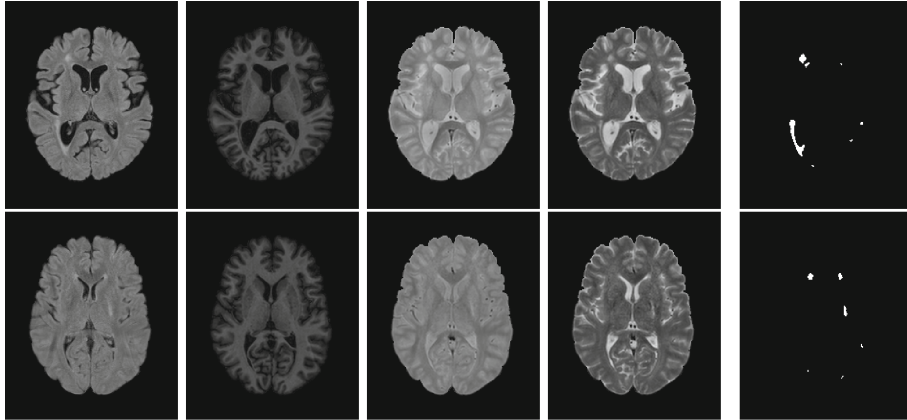


Fig. 7. Main challenge results. *Columns, left to right:* FLAIR, MPRAGE, PD and T2 scan together with the computed segmentation at Slice 80. *Rows:* best (*top*) and worst (*bottom*) segmentation (baseline scan of patient 4 and 7 respectively), with respect to the challenge score computed on both raters' segmentations.

4 Discussion

We encountered expected as well as odd behavior in our exploratory study. Contrary to what has been advised for in the literature [12], dropping information from the state weights results in better regularization, as the Dice coefficients in Table 1B indicate. This behavior could be due to the fact that we only ignore part of the previous state per iteration and channel. Interestingly though, a

combination of DC on both input and state produces worse results, even with a reduced drop rate. As dropout tends to prolong training, further experiments with longer training times might shed light on this effect. Another surprising result is the inability of BN to surpass baseline Dice scores in our preliminary tests in Table 1, in all variations we tested. Due to the correlation in our mini-batch of one and the varying weights in the case of the running average, the assumption does not hold that the statistics of our mini-batch are similar to the global statistics. Residual learning between MD-GRU layers seems to contribute to the overall improvement. Surprisingly, neither concatenating the C-GRU nor placing the reset gate as in the original GRU did result in an improved Dice.

The high pass filtering as preprocessing step proved to be fruitful, especially in the setting where we only trained for 3000 iterations, where leaving it out resulted in no segmentation at all. Using only original data, a visible tendency towards lesions could be found, but with probabilities well below 0.5. The main reason why this step is so important can be seen in Fig. 3, where values of the filtered image lie mostly around zero and in the original scan around two. All the weights of our network are initialized to handle data from a standard normal distribution. Inside the brain, filtering the original image would result in sums far away from zero. Using a hyperbolic tangent or sigmoid function on such a result will return a value close to 1 and hence a very flat gradient, which will not be able to help adjust the weights to correct for this in a fast manner.

Selective sampling and random deformation succeeded to be the most important improvements, which is easily explainable with the huge class imbalance present in our data and the low amount of training data. As the crossvalidation shows, all of the selected techniques resulted in overall better scores except for the HD in selective sampling, which is likely due to a higher probability of producing outliers when oversampling the lesion class.

By achieving rank 1 in the actual challenge, we show that our proposed method is at the state of the art. Unfortunately, none of the listed results in Table 3 have been published yet, as already mentioned. The highest Dice score in the top 5 was achieved using exactly the same MD-GRU network as in its original publication [1] and training it for 40 000 iterations. Since we only trained our network for 10 000 iterations and showed superior performance over the original setup in our evaluation, we believe that an even higher score is possible by training for a longer time.

MD-GRUs allow for any number of dimensions in the data, hence it would also be possible to use the actual 4d data from the challenge, with the new dimension being the screening number. Using 4d data could pose a number of problems though, for instance the reduced spatial resolution that can be fed to the network per training iteration due to the memory constraints and the low number of screenings that are available. Further research might be necessary to determine, if a suitable trade-off between spatial resolution and temporal information exists.

In conclusion, the following four modifications can drastically improve the accuracy of lesion segmentation in terms of Dice, HD and AVD with the

MD-GRU: Selective sampling speeds up training drastically, since most of the data can be labeled safely as background. DC on the state does a better job in regularization than on the input. Random deformation prevents the model from overfitting. Finally, residual learning in between MD-GRUs might shorten training time by simplifying the estimation task.

References

1. Andermatt, S., Pezold, S., Cattin, P.: Multi-dimensional gated recurrent units for the segmentation of biomedical 3D-data. In: Carneiro, G., et al. (eds.) LABELS/DLMIA-2016. LNCS, vol. 10008, pp. 142–151. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46976-8_15
2. Carass, A., Roy, S., Jog, A., Cuzzocreo, J.L., Magrath, E., Gherman, A., But-ton, J., Nguyen, J., Prados, F., Sudre, C.H., Jorge Cardoso, M., Cawley, N., Ciccarelli, O., Wheeler-Kingshott, C.A.M., Ourselin, S., Catanese, L., Deshpande, H., Maurel, P., Commowick, O., Barillot, C., Tomas-Fernandez, X., Warfield, S.K., Vaidya, S., Chunduru, A., Muthuganapathy, R., Krishnamurthi, G., Jesson, A., Arbel, T., Maier, O., Handels, H., Ithme, L.O., Unay, D., Jain, S., Sima, D.M., Smeets, D., Ghafoorian, M., Platel, B., Birenbaum, A., Greenspan, H., Bazin, P.L., Calabresi, P.A., Crainiceanu, C.M., Ellingsen, L.M., Reich, D.S., Prince, J.L., Pham, D.L.: Longitudinal multiple sclerosis lesion segmentation: resource and challenge. *NeuroImage* **148**, 77–102 (2017)
3. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. [arXiv:1406.1078](https://arxiv.org/abs/1406.1078) [cs, stat], June 2014
4. Cooijmans, T., Ballas, N., Laurent, C., Gülçehre, Ç., Courville, A.: Recurrent batch normalization. [arXiv:1603.09025](https://arxiv.org/abs/1603.09025) [cs], March 2016
5. Filippi, M., Horsfield, M.A., Bressi, S., Martinelli, V., Baratti, C., Reganati, P., Campi, A., Miller, D.H., Comi, G.: Intra- and inter-observer agreement of brain MRI lesion volume measurements in multiple sclerosis. *Brain* **118**(6), 1593–1600 (1995)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. [arXiv:1512.03385](https://arxiv.org/abs/1512.03385) [cs], December 2015
7. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. [arXiv:1502.03167](https://arxiv.org/abs/1502.03167) [cs], February 2015
8. Liao, Q., Poggio, T.: Bridging the gaps between residual learning, recurrent neural networks and visual cortex. [arXiv:1604.03640](https://arxiv.org/abs/1604.03640) [cs], April 2016
9. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **45**(11), 2673–2681 (1997)
10. Stollenga, M.F., Byeon, W., Liwicki, M., Schmidhuber, J.: Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 28, pp. 2998–3006. Curran Associates, Inc. (2015)
11. Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R.: Regularization of neural networks using dropconnect. In: *Proceedings of the 30th International Conference on Machine Learning (ICML-2013)*, pp. 1058–1066 (2013)
12. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. [arXiv:1409.2329](https://arxiv.org/abs/1409.2329) [cs], September 2014

Chapter 7

Automatic Landmark Localization

Introduction

In a collaboration with Simon Pezold, we aimed at automating the localization of the medullopontine sulcus. The landmark is used in a spinal cord segmentation work-flow, and was selected manually so far. We adapted the MD-GRU setup, such that also a reduction of spatial resolution was possible and could hence formulate a network which directly estimates a location. Instead of regressing a coordinate, we rely on classification, and propose two measures to overcome limitations of this discrete binning. The work was divided such that the development of the network and handling of experiments was performed by Simon Andermatt and the data loading routine and literature review were written by Simon Pezold, whereas the rest of the work load was shared.

Technical Report This paper has been submitted to `arxiv.org`¹.

¹This technical report is hosted at <https://arxiv.org/abs/1708.02766>

Multi-dimensional Gated Recurrent Units for Automated Anatomical Landmark Localization

Simon Andermatt^{*,1}, Simon Pezold^{*,1}, Michael Amann^{2,3,4}, and
Philippe C. Cattin¹

¹Department of Biomedical Engineering, University of Basel, Allschwil, Switzerland

²Department of Neurology, University Hospital Basel, Basel, Switzerland

³Department of Radiology, University Hospital Basel, Basel, Switzerland

⁴MIAC AG, Basel, Switzerland

^{*}S. Andermatt and S. Pezold contributed equally.

Abstract. We present an automated method for localizing an anatomical landmark in three-dimensional medical images. The method combines two recurrent neural networks in a coarse-to-fine approach: The first network determines a candidate neighborhood by analyzing the complete given image volume. The second network localizes the actual landmark precisely and accurately in the candidate neighborhood. Both networks take advantage of multi-dimensional gated recurrent units in their main layers, which allow for high model complexity with a comparatively small set of parameters. We localize the medullopontine sulcus in 3D magnetic resonance images of the head and neck. We show that the proposed approach outperforms similar localization techniques both in terms of mean distance in millimeters and voxels w.r.t. manual labelings of the data. With a mean localization error of 1.7 mm, the proposed approach performs on par with neurological experts, as we demonstrate in an inter-rater comparison.

1 Introduction

Localizing anatomical landmarks is a common task in many medical applications. Finding matching anatomical points in images may be necessary for seeding a segmentation algorithm, for registration problems, or for providing points of reference for quantitative measurements. Although finding landmarks in volumetric images is error-prone and time-consuming, the task is often still carried out manually. Using a fully automated approach mitigates the inter and intra-rater variability through an objective and efficient process without manual interference. Therefore, many automated localization methods have been proposed, with varying degrees of robustness, reliability, and generalization potential. Some of the methods, such as Bhanu Prakash et al. [2] or Elattar et al. [3], use very basic image processing techniques, but many others rely on concepts from machine learning: for example, for localizing landmarks in the brain, Guerrero et al. [6] use manifold learning and O’Neil et al. [9] use random forests; for cardiac landmark localization, Karavides et al. [7] use Adaboost and Lu and Jolly [8] use

probabilistic boosting trees; Xue et al. [11] use boosting for localizing landmarks on the knee joint. For a recent overview, also see Zhou et al. [15].

In recent years, ground-breaking advancements using neural networks have been achieved in various domains, allowing for automatic learning of discriminative features for the problem at hand and avoiding the need for manually designed (often called handcrafted) features. Consequently, these techniques have also found their way into landmark localization. Examples are Zheng et al. [14], who use two neural networks successively to localize the carotid bifurcation in 3D CT images, Ghesu et al. [4], who propose a so-called artificial agent for localizing various anatomical landmarks in 2D and 3D images of different modalities, and Yang et al. [12], who apply convolutional neural networks for landmark localization on the femur in MR images.

Existing approaches based on convolutional neural networks (CNNs) are capable of detecting very delicate structure, yet are limited to the local neighborhood of the filters used in each layer of the network. Using a recurrent neural network (RNN) for this task allows for flexible feature relationships of varying length and scale. This is especially useful given a localization task, where the surrounding tissues structure can take a number of different shapes and sizes. Tackling volumetric data with RNNs for *segmentation* has been recently demonstrated by Andermatt et al. [1] with multi-dimensional gated recurrent units (MD-GRUs). To our knowledge, neither multi-dimensional RNN nor MD-GRUs have been applied to the task of *landmark localization* so far.

In this paper, we propose to apply MD-GRUs in a two-stage approach to the task of anatomical landmark localization. In the first stage, the anatomical region of interest is roughly located in the given image volume. We then determine the actual landmark coordinate in a subvolume in the second stage. We apply the proposed method to 3D MR images of the head and neck, in which we locate the medullopontine sulcus, and compare the found coordinates to those of manual labels. Our results from an interrater comparison suggest that the proposed method cannot be distinguished from a clinical expert.

2 Methods

For the accurate localization of landmarks, we propose to use two separate localization networks of similar structure, to both accelerate the process and allow for a decently complex network. Both localization networks work on the same number of voxels – in our case we fixed it to 64^3 voxels – and find the coordinate in said volume which lies closest to the true landmark. The first network is provided data subsampled to such a degree, that the full original volume can be represented inside of it. The network will then approximate a location, which will in turn be used to sample a subvolume at the original resolution from the image data around the found location. In our case, the first network is provided with 4-fold subsampled data and the second processes data at the original resolution, centered at the location which was found by the first network.

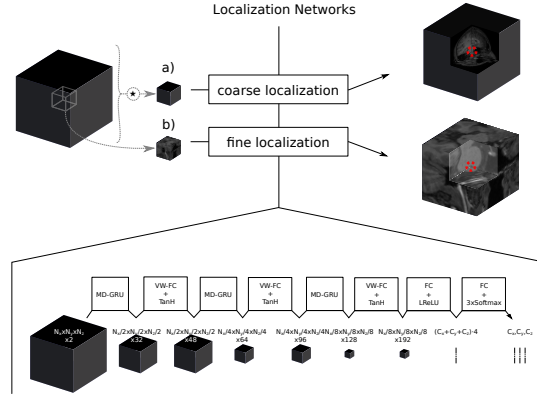


Fig. 1. Localization network. a) Coarse approximation of landmark coordinates in sub-sampled low resolution representation of full data. b) Fine approximation of landmark coordinates in extracted window around detected coarse location in a second localization network. Both networks use the architecture depicted at the bottom.

Subsampling MD-GRU Layer We propose to adapt the MD-GRU layer [1], which was introduced to handle segmentation problems, to the application of landmark localization. In order to do so, we implement the ability to subsample at each MD-GRU layer and hence at each convolutional gated recurrent unit (C-GRU) which it consists of. This effectively reduces the spatial problem size, allowing a multi-resolution processing approach. We adjust the original C-GRU equations as follows:

$$f^j(t, \alpha, \beta) = \sum_i x_t^i \star \alpha^{i,j} + \beta^j, \quad g^j(t, \alpha) = \sum_k h_{t-1}^k \star \alpha^{k,j}, \quad (1)$$

$$r_t^j = \sigma(f^j(t, w_r, b_r) + g^j(t, u_r)), \quad z_t^j = \sigma(f^j(t, w_z, b_z) + g^j(t, u_z)), \quad (2)$$

$$\tilde{h}_t^j = \phi(f^j(t, w, b) + r_t^j \odot g^j(t, u)), \quad h_t^j = z_t^j \odot h_{t-1}^j + (1 - z_t^j) \odot \tilde{h}_t^j, \quad (3)$$

where x_t^i , h_t^i denote the input and state of the C-GRU at time t , and i, j, k denote the respective channels. The operator \odot denotes elementwise multiplication, as in [1]. Variables u , w , and b are trainable weights. We call \tilde{h} in Eq. (3) the proposal and r and z in Eqs. (2) the reset and update gate.

We accomplish subsampling by introducing strided convolutions, which are denoted as \star in Eq. (1). The size of the state as well as of all the gates and the proposal will be reduced by the factor of the chosen stride S per spatial dimension. Each C-GRUs' output is then subjected to one-dimensional average pooling, compressing the time dimension by stride S . The sum of all d compressed

C-GRU results \hat{h} yields the MD-GRU output H :

$$H^j = \sum_d \hat{h}^j, \quad \hat{h}_{t'}^j = \frac{1}{S} \sum_{s=0}^{S-1} h_{St'+s}^j. \quad (4)$$

Localization Network At the core, we use the same localization network for all experiments. We use three subsequent compositions of a subsampling MD-GRU layer, a voxelwise fully connected layer, and a tanh activation function. The subsampling MD-GRU layers are provided with 32, 64, and 128 channels, respectively. All of them use strides of 2 along spatial dimensions, the volume is hence subsampled 8-fold at each composition. We use DropConnect [10] with a drop rate of 0.5 on the input convolution filters of both gates r^j , z^j and the proposal \tilde{h} . The voxelwise fully connected layers are realized through convolution layers with spatial filters of 1^3 , with 48, 96, and 192 channels each.

The resulting subvolume is of size $N_x/8 \times N_y/8 \times N_z/8$, given the input shape was $(N_x \times N_y \times N_z)$. The subvolume is reshaped into a vector, in which we process each coordinate by two fully connected layers of $(C_x + C_y + C_z) \cdot 4$ and $(C_x + C_y + C_z)$ layers, which are connected through a leaky rectifying unit defined as $\text{lrelu}(x) = \max\{0.01x, x\}$. The resulting vector is split into three separate vectors of sizes C_x , C_y , and C_z , where C gives the number of possible coordinate positions along the respective dimension. These are then fed into individual softmax activation functions to estimate the probabilities for each coordinate in each vector. We use the sum of all cross entropy losses as loss function for the entire network. Figure 1 shows an overview of the network architecture.

Subsampling In the first stage, we use a strided convolution on the input to match the localization networks input resolution. We pad the input, such that the shape of the volume is a multiple of the required shape for the localization network. In our case, we padded the data to 256^3 and used strides S of 4 with a filter size of $S \cdot 2 + 1$ and 16 channels for the convolution layer.

Superresolution Our method, as explained so far, is restricted to voxel coordinates, since we estimate with our method discrete instead of continuous coordinates. In the following, we explain two extensions to our idea to yield superresolution results.

The first extension takes advantage of the coordinate resolution-independent formulation in the *Localization Network* paragraph above. Instead of estimating as many classes for each of the three coordinates as there are voxels in the respective dimension in the volume, we estimate n times the amount. This allows us to estimate values which are $1/n$ voxels apart and hence allow for a more fine-grained localization. In our experiments, we use $n = 4$ resulting in 256 classes.

Our second idea exploits neighborhood information in our coordinate probability vectors by fitting a parabola to the largest probability and its two neighbors per coordinate. The maxima of these functions can then be interpreted as our

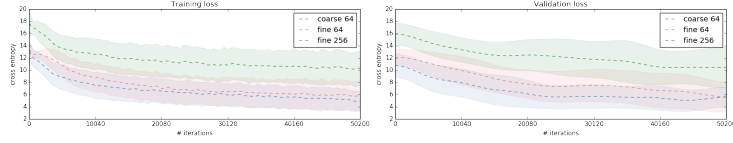


Fig. 2. Cross entropy loss. Mean \pm one standard deviation on training and validation set for the 3 trained networks, smoothed using a gaussian for visualization.

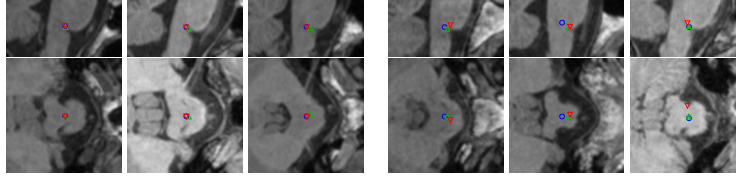


Fig. 3. Localization results for rater 1 (red ∇), rater 2 (green \triangle), and proposed method (blue \circ). Shown are the best three (left) and worst three (right) localizations of the proposed method wrt. rater 1, both in sagittal (top) and transverse (bottom) view.

coordinate location. This allows for an even finer localization, but is based and hence limited on the chosen number of coordinate probabilities.

Optimization We trained each localization network together with their sub-sampling addition individually. All networks were trained for a total of 50 epochs, where one epoch comprised one random sample from each training subject, which led to a total of 50 200 iterations. We used AdaDelta [13] with a learning rate of 0.001. We initialized all weights of the convolutions with the method of Glorot and Bengio [5], the biases with zero and the fully connected layers at the end of the localization network with random values from $[-\sqrt{3}/N_i, +\sqrt{3}/N_i]$, where N_i is the number of input units. For the first network, we sampled from the center of the padded volume with a random offset in the range of $[-100, 100]$ voxels per coordinate; for the second network, we just required that the training landmark was within the volume. The training loss is visualized in Fig. 2.

For preprocessing, we apply a high-pass filter on the input, the results of which we use together with the original data as input to our networks. Additionally, we normalize to zero mean and a standard deviation of one for each of the input volumes. Apart from this, no preprocessing is required.

3 Results

To evaluate the proposed approach, we located the medullopontine sulcus, a distinct cavity in the brainstem, in MR images of the head and neck (see Fig. 3).

Table 1. Localization accuracy and precision. a) Localization error on the test set when using only the first network (*top row*) and both networks with a varying number of coordinate classes, with or without parabola fitting (*bottom row*: proposed combination); b) localization error on the test set in comparison to two human raters; c) localization errors reported in the literature.

a)	Error [mm]			b)	Error [mm]		
	Median	Mean	Std.		Median	Mean	Std.
Coarse localization	4.83	5.02	2.22	Rater 1 vs. rater 2	1.39	1.59	0.98
Fine, 64 classes	1.74	1.97	1.02	Proposed vs. rater 1	1.40	1.69	1.02
Fine+parab., 64 cl.	1.77	1.89	0.98	Proposed vs. rater 2	1.65	1.73	0.87
Fine, 256 classes	1.47	1.72	1.03	Proposed vs. both	1.50	1.71	0.95
Fine+parab., 256 cl.	1.40	1.69	1.02				

c)	Error [mm]			Voxel size [mm ³]	Target landmark
	Median	Mean	Std.		
Proposed	1.50	1.71	0.95	$1.00 \times 1.00 \times 1.00$	medullopontine sulcus
Zheng et al. [14]	1.21	2.64	4.98	$0.46 \times 0.46 \times 0.50$	carotid bifurcation
Ghesu et al. [4]	0.8	1.8	2.9	$1.00 \times 1.00 \times 1.00$	carotid bifurcation
Yang et al. [12]	—	4.13	1.70	$0.37 \times 0.37 \times 0.70$	femoral medial distal point
Xue et al. [11]	—	1.41	0.91	$0.3 \times 0.3 \times [0.6, 3]$	knee joint (23 landmarks)
Guerrero et al. [6]	—	0.45	0.22	—	anterior commissure

Images were acquired with a T1-weighted MPRAGE sequence, having a resolution of 1 mm³ and a size between 160 × 240 × 256 voxels and 192 × 256 × 256 voxels. Altogether, we had 1218 images of 265 subjects, with a median number of 5 images per subject (minimum: 1, maximum: 8), which we randomly assigned to a training set (1004 images of 213 subjects), a validation set (114 images of 26 subjects), and a test set (100 images of 26 subjects), making sure that all images of each subject were assigned to the same set.

For training and evaluation of the localization, we used manual labels of the landmark. These labels were provided by clinical expert raters who placed them on a graphical user interface enabling them to zoom in and out of the imaged volumes as necessary. To allow for interrater comparisons, we had two raters place the landmark in all images of the test set.

Training 50 epochs for the coarse and fine networks took around 41 and 34 hours, respectively. Testing, on the other hand, requires less than 2 seconds for either network, resulting in a total of around 3–4 seconds for localization. Using our extension of estimating 256 class probabilities instead of 64 per coordinate requires only 2.5 hours more training time and took around 2.5 seconds per volume for testing, which results in around 4 seconds in total for localization.

Figure 3 shows our three best and worst localization results. Note that our largest error (rightmost column in Fig. 3) is actually produced by a mislabeling of a clinical expert, as can be seen by the off-center position of the red marker.

Table 1a shows the localization errors when using only the first network as compared to using both. The second network increases the localization accuracy notably, as does using more coordinate classes and fitting a parabola.

Table 1b shows the results from comparing both human raters with the proposed approach. The listed values indicate that our approach almost reaches human performance: comparing our results to those of a human rater produces approximately the same error as two human raters compared to each other.

Table 1c shows results for landmark localization reported in the literature.

4 Discussion and Conclusion

Our results, as listed in Table 1c, appear competitive: compared to other neural network approaches [4,12,14], mean error and standard deviation are better in terms of millimeters and voxels. When comparing to Xue et al. [11], one has to keep in mind their notably higher in-plane resolution. While Guerrero et al. [6] achieve higher accuracy and precision, a comparison appears difficult: apart from not stating the voxel size, their method requires images with similar field of view, which cannot be guaranteed in our case, as parts of our images are centered on the neck while others are centered on the head. In any case, caution has to be taken when comparing these results: on the one hand, evaluated anatomical landmarks, imaging modalities, and image resolutions differ. On the other hand, our interrater comparison (recall Table 1b) suggests that there is a lower bound for the achievable accuracy, which might be well above a given image resolution and might depend on the particular anatomical landmark. Determining the limit of actually achievable accuracy of our method would require evaluating data with lower interrater variability. The results of Xue et al. [11] allow a similar conclusion, in that their method’s error is similar to the error from their interrater comparison, as well. Unfortunately, the other authors do not provide interrater comparisons.

We have shown two ideas that improved our localization results. The combination of both even surpassed the accuracy of each of them applied separately. Considering interrater variability, we are still slightly less accurate than a human rater. We think that this is partly based on the discrete probability distribution and our sampling technique when training the algorithm. We randomly sampled subvolumes using integer coordinates during training since this process does not require interpolation. But this also means that each training sample could only get mapped on a subset of all possible coordinate classes.

Conclusion We have shown that the localization of the medullopontine sulcus is successfully possible using our proposed automated technique, which adapts MD-GRUs to the task of landmark localization. We introduced a number of improvements, which all led to even more accurate results without significantly increasing the training time. Future work will focus on evaluating our localization approach on multiple anatomical landmarks in different imaging modalities.

References

1. Andermatt, S., Pezold, S., Cattin, P.: Multi-dimensional Gated Recurrent Units for the Segmentation of Biomedical 3D-Data. In: International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis. pp. 142–151. Springer (2016)
2. Bhanu Prakash, K.N., Hu, Q., Aziz, A., Nowinski, W.L.: Rapid and Automatic Localization of the Anterior and Posterior Commissure Point Landmarks in MR Volumetric Neuroimages. *Academic Radiology* 13(1), 36–54 (Jan 2006)
3. Elattar, M., Wiegierinck, E., van Kesteren, F., Dubois, L., Planken, N., Vanbavel, E., Baan, J., Marquering, H.: Automatic aortic root landmark detection in CTA images for preprocedural planning of transcatheter aortic valve implantation. *The International Journal of Cardiovascular Imaging* 32(3), 501–511 (Mar 2016)
4. Ghesu, F.C., Georgescu, B., Mansi, T., Neumann, D., Hornegger, J., Comaniciu, D.: An Artificial Agent for Anatomical Landmark Detection in Medical Images. In: MICCAI 2016. pp. 229–237. Springer, Cham (Oct 2016)
5. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: *Aistats*. vol. 9, pp. 249–256 (2010)
6. Guerrero, R., Wolz, R., Rueckert, D.: Laplacian Eigenmaps Manifold Learning for Landmark Localization in Brain MR Images. In: MICCAI 2011. pp. 566–573. Springer, Berlin, Heidelberg (Sep 2011)
7. Karavides, T., Leung, K.Y.E., Paclik, P., Hendriks, E.A., Bosch, J.G.: Database guided detection of anatomical landmark points in 3D images of the heart. In: 2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro. pp. 1089–1092 (Apr 2010)
8. Lu, X., Jolly, M.P.: Discriminative Context Modeling Using Auxiliary Markers for LV Landmark Detection from a Single MR Image. In: *Statistical Atlases and Computational Models of the Heart. Imaging and Modelling Challenges*. pp. 105–114. Springer, Berlin, Heidelberg (Oct 2012)
9. O’Neil, A., Dabbah, M., Poole, I.: Cross-Modality Anatomical Landmark Detection Using Histograms of Unsigned Gradient Orientations and Atlas Location Autocontext. In: *Machine Learning in Medical Imaging*. pp. 139–146. Springer, Cham (Oct 2016)
10. Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R.: Regularization of neural networks using dropconnect. In: *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. pp. 1058–1066 (2013)
11. Xue, N., Doellinger, M., Ho, C.P., Surowiec, R.K., Schwarz, R.: Automatic detection of anatomical landmarks on the knee joint using MRI data. *Journal of Magnetic Resonance Imaging* 41(1), 183–192 (Jan 2015)
12. Yang, D., Zhang, S., Yan, Z., Tan, C., Li, K., Metaxas, D.: Automated anatomical landmark detection on distal femur surface using convolutional neural network. In: 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI). pp. 17–21 (Apr 2015)
13. Zeiler, M.D.: ADADELTA: An Adaptive Learning Rate Method. arXiv:1212.5701 [cs] (Dec 2012)
14. Zheng, Y., Liu, D., Georgescu, B., Nguyen, H., Comaniciu, D.: 3D Deep Learning for Efficient and Robust Landmark Detection in Volumetric Data. In: MICCAI 2015. pp. 565–572. Springer, Cham (Oct 2015)
15. Zhou, S.K.: Discriminative anatomy detection: Classification vs regression. *Pattern Recognition Letters* 43, 25–38 (Jul 2014)

Chapter 8

Weakly-Supervised Pathology Segmentation

Introduction

The previously discussed, supervised approaches require fully annotated data. For the task of semantic segmentation, voxel-wise and usually hand-labeled segmentation maps for each training sample are required. As each new lesion class or set of MR sequences requires a new training set, this approach is not very flexible. Image data which are simply classified into categories, on the other hand, are produced on a daily basis in medical routine. Even hand labeling such a dataset takes orders of magnitude less work compared to labeling each voxel by hand. We propose to use only one binary, image-level label stating if the respective image contains pathology of a given kind or not. Using generative adversarial networks and variational autoencoders, we formulate two transformations, one which renders healthy data pathological by selecting a region of interest to be changed into pathological tissue, and one which segments pathological tissue and creates a healthy inpainting for a pathological region. We show, that segmentations created with our method are meaningful and, albeit not on the level of supervised methods, produce useful segmentations when applied to brain tumor segmentation.

Publication This paper has been submitted to `arxiv.org`¹ and to the 4th workshop on *Brain-Lesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries* (Brainles) which will be held at MICCAI in 2018.

¹This technical report is hosted at <https://arxiv.org/abs/1805.10344>

Pathology Segmentation using Distributional Differences to Images of Healthy Origin

Simon Andermatt, Antal Horváth, Simon Pezold, and Philippe Cattin

Department of Biomedical Engineering, University of Basel, Allschwil, Switzerland

Abstract. We present a method to model pathologies in medical data, trained on data labelled on the image level as healthy or containing a visual defect. Our model not only allows us to create pixelwise semantic segmentations, it is also able to create inpaintings for the segmentations to render the pathological image healthy. Furthermore, we can draw new unseen pathology samples from this model based on the distribution in the data. We show quantitatively, that our method is able to segment pathologies with a surprising accuracy and show qualitative results of both the segmentations and inpaintings. A comparison with a supervised segmentation method indicates, that the accuracy of our proposed weakly-supervised segmentation is nevertheless quite close.

1 Introduction

Supervised segmentation in medical image analysis is an almost solved problem, where methodological improvements have a marginal effect on accuracy. Such methods depend on a large annotated training corpus, where pixelwise labels have to be provided by medical experts. In practice, such data are expensive to gather. In contrast, weakly labelled data, such as images showing a certain disease are easily obtainable, since they are created on a daily basis in medical practice. We want to take advantage of these data for pathology segmentation, by providing a means to finding the difference between healthy and pathological data distributions. We present a weakly supervised framework capable of pixelwise segmentation as well as generating samples from the pathology distribution.

Our idea is inspired by CycleGAN [10], a recently proposed solution for unpaired image to image translation, where the combination of domain-specific generative adversarial networks (GANs) and so-called cycle consistency allow for robust translation. We call our adaptation PathoGAN and count the following contributions: We formulate a model capable of segmentation based on a single label per training sample. We simultaneously train two generative models, able to generate inpaintings at a localized region of interest to transform an image from one domain to the other. We are able to sample healthy brains as well as sample possible pathologies for a given brain. Furthermore, our method enforces domain-specific information to be encoded outside of the image, which omits adversarial “noise” common to CycleGAN [4] to some degree.

We show the performance of our implementation on 2d slices of the training data of the Brain Tumor Segmentation Challenge 2017 [8,2] and compare our segmentation performance to a supervised segmentation technique [1].

CycleGAN has been previously used to segment by transferring to another target modality, where segmentation maps are available (e.g. [9]), or applied to generate training from cheaply generated synthetic labelmaps [5]. Using a Wasserstein GAN, another method directly estimates an additive visual attribution map [3]. To our knowledge, there has not been a method that jointly learns to segment on one medical imaging modality using only image-level labels and generate new data using GANs for both healthy and pathological cases.

2 Methods

2.1 Problem Statement

We assume two image domains \mathcal{A} and \mathcal{B} , where the former contains only images of healthy subjects and the latter consists of images showing a specific pathology. We seek to approximate the functions $G_{\mathcal{A}}$ and $G_{\mathcal{B}}$ that perform the mappings $(x_{\mathcal{A}}, \delta_{\mathcal{B}}) \mapsto \hat{x}_{\mathcal{B}}$ and $(x_{\mathcal{B}}, \delta_{\mathcal{A}}) \mapsto \hat{x}_{\mathcal{A}}$, where $x_{\mathcal{A}}, \hat{x}_{\mathcal{A}} \in \mathcal{A}$ and $x_{\mathcal{B}}, \hat{x}_{\mathcal{B}} \in \mathcal{B}$. Vectors $\delta_{\mathcal{B}}$ and $\delta_{\mathcal{A}}$ encode the missing target image information (e.g. the pathology):

$$\hat{x}_{\mathcal{B}} = G_{\mathcal{A}}(x_{\mathcal{A}}, \delta_{\mathcal{B}}), \quad \hat{x}_{\mathcal{A}} = G_{\mathcal{B}}(x_{\mathcal{B}}, \delta_{\mathcal{A}}). \quad (1)$$

In the remaining paper, we use X and Y as placeholders for \mathcal{A} and \mathcal{B} or \mathcal{B} and \mathcal{A} to overcome redundancy due to symmetrical components. We encourage $G_{\mathcal{A}}, G_{\mathcal{B}}$ to produce results, such that the mappings are realistic (2), bijective (3), specific (4) and that only the affected part in the image is modified (5):

$$G_X(x_X, \delta_Y) \sim Y, \quad (2) \quad G_Y(x_X, 0) \approx x_X, \quad (4)$$

$$G_Y(G_X(x_X, \delta_Y), \delta_X) \approx x_X, \quad (3) \quad G_X \approx \arg \min |x_X - G_X(x_X, \delta_Y)|. \quad (5)$$

2.2 Model Topology

To fulfill Eqs. (2–5), we adopt the main setup and objective from CycleGAN: we employ two discriminators, $D_{\mathcal{A}}$ and $D_{\mathcal{B}}$ together with generators $G_{\mathcal{A}}$ and $G_{\mathcal{B}}$ to perform the translation from domain \mathcal{A} to \mathcal{B} and vice versa, formulating two generative adversarial networks (GANs) [6]. In both directions, the respective discriminator is trained to distinguish a real image from the output of the generator, whereas the generator is incentivized to generate samples that fool the discriminator.

Residual Generator In order to segment pathologies, we seek to only modify a certain part of the image. In contrast to CycleGAN, we model the transformation G from one domain to the other as a residual or inpainting p which is exchanged with part l of the original image. We achieve this by letting generator G_X directly

estimate $n + 1$ featuremaps r_X , where n is the number of image channels used. We obtain labelmap l_X and inpaintings p_X , activating $r_X^{(0)}$ with a sigmoid and each $r_X^{(i)}$ with a tanh activation:

$$l_X = S\left(r_X^{(0)} + \epsilon\right), \quad p_X^{(i-1)} = \tanh\left(r_X^{(i)}\right), \quad (6)$$

where $S(y) = \frac{1}{1+e^{-y}}$ and $i > 0$. With $\epsilon \sim \mathcal{N}(0, I)$, we turn $r_X^{(0)} + \epsilon$ into samples from $\mathcal{N}(r_X^{(0)}, I)$ using the reparameterization trick [7]. This allows reliable calculations of l_X only for large absolute values of $r_X^{(0)}$, forcing l_X to be binary and intensity information to be encoded in the inpaintings. We set ϵ to zero during testing. From l_X and p_X we compute the translated result \hat{x}_Y , supposedly in domain Y now:

$$\hat{x}_Y = l_X \odot p_X + (1 - l_X) \odot x_X.$$

In the following, we detail the computation of r for the two possible translation directions. Both translation pathways are visualized in Fig. 1.

$\mathcal{A} \rightarrow \mathcal{B}$ To map from healthy to pathological data, we estimate $r_{\mathcal{A}}$ (and thus $l_{\mathcal{A}}, p_{\mathcal{A}}$) using a variational autoencoder (VAE) [7]. First, we employ encoders $\Gamma_{\mathcal{A}}$ and $\Delta_{\mathcal{B}}$ to encode anatomical information around and inside the pathological region:

$$\gamma_{\mathcal{A}} = \Gamma(x_{\mathcal{A}}), \quad \delta_{\mathcal{B}} = \Delta(l'_{\mathcal{B}} \odot x'_{\mathcal{B}}),$$

where $x_{\mathcal{A}}$ is our healthy image, $l'_{\mathcal{B}}$ and $x'_{\mathcal{B}}$ are the labelmap and pathological image of the previous transformation and $\delta_{\mathcal{B}}, \gamma_{\mathcal{A}} \sim \mathcal{N}(0, I)$. If $l'_{\mathcal{B}}$ and $x'_{\mathcal{B}}$ are not available because $x_{\mathcal{A}}$ is a real healthy image, we simply sample δ_y . Finally, a decoder $E_{\mathcal{A}}$ is applied to $\gamma_{\mathcal{A}}$ and $\delta_{\mathcal{B}}$:

$$r_{\mathcal{A}} = E_{\mathcal{A}}(\gamma_{\mathcal{A}}, \delta_{\mathcal{B}}).$$

$\mathcal{B} \rightarrow \mathcal{A}$ To generate healthy samples from pathological images, we use a generator $E_{\mathcal{B}}$ directly on the input as introduced in [10] and estimate r directly:

$$r_{\mathcal{B}} = E_{\mathcal{B}}(x_{\mathcal{B}}).$$

Here, we omit $\delta_{\mathcal{A}}$, since the location and appearance of missing healthy tissue can be inferred from $x_{\mathcal{B}}$. We also omit using an encoding bottleneck due to possible information loss and less accurate segmentation.

2.3 Objective

To train this model, a number of different loss terms are necessary. In the following, we explain the individual components using $\hat{\cdot}$ and $\tilde{\cdot}$ to denote results from the translated and reconstructed images respectively (e.g. mapping x_X into Y results in \hat{x}_Y , translating it back results in \tilde{x}_X). We use λ_{\cdot} to weight the contribution of different loss terms.

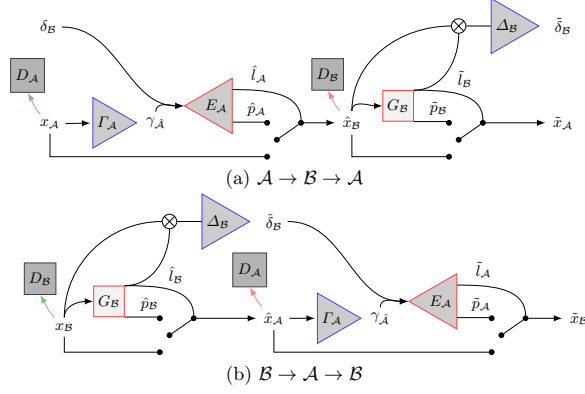


Fig. 1. Proposed architecture: *top to bottom*: directions $\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{A}$ and $\mathcal{B} \rightarrow \mathcal{A} \rightarrow \mathcal{B}$; x_A, x_B are samples from the two data distributions \mathcal{A}, \mathcal{B} and $\delta_B \sim \mathcal{N}(0, I)$. Red and blue triangles depict decoder and encoder networks. A red square illustrates a simple generators. Δ_B and Γ_A encode features inside and outside the pathological region. Δ_B, Γ_A and E_A form a variational autoencoder. For G_B , information about the missing healthy structure is completely inferred from the surroundings.

CycleGAN As in CycleGAN [10], we formulate a least squares proxy GAN loss, which we minimize with respect to G_X and maximize with respect to D_Y :

$$\mathcal{L}_{\text{GAN}}(D_Y, G_X, x_X, x_Y) = \mathbb{E}[(D_Y(x_Y))^2] + \mathbb{E}[(1 - D_Y(G_X(x_X, \delta_Y)))^2], \quad (7)$$

Likewise, to make mappings reversible, we add the cycle consistency loss:

$$\mathcal{L}_{\text{CC}}(G_X, G_Y, x_X) = \lambda_{\text{CC}} \|G_Y(G_X(x_X, \delta_Y), \delta_X) - x_X\|_1. \quad (8)$$

\mathcal{L}_{GAN} and \mathcal{L}_{CC} encourage the properties defined in Eqs. (2) and (3).

Variational Autoencoder A variational autoencoder (VAE) is trained by minimizing the KL-divergence of the distribution $q(z|x)$ of encoding z to some assumed distribution and the expected reconstruction error $\log p(x|z)$, where x is the data. In contrast to a classical VAE, we use two distinct encoding vectors γ_A and δ_B , encoding the healthy and the pathological part, and produce two separate results, the labelmap l_A and the inpainting p_A . We directly calculate the KL-divergence for our two encodings:

$$\mathcal{L}_{\text{KL}}(G_A, G_B, x_A, x_B) = \text{KL}[q(\gamma_A|x_A)|\mathcal{N}(0, I)] + \text{KL}[q(\delta_B|x_B, \hat{l}_B)|\mathcal{N}(0, I)]. \quad (9)$$

For the expected reconstruction error, we assume that l and p follow approximately a Bernoulli and a Gaussian distribution ($\mathcal{N}(\mu, I)$). We selectively penalize

the responsible encoding, by using separate loss functions for the residual region and the rest. Unfortunately, we only ever have access to the ground truth of one of these regions, since we do not use paired data. We solve this, by using the relevant application in the network, where individual ground truths are available, to calculate the approximation of the marginal likelihood lower bound:

$$\mathcal{L}_{\text{VAE}}(G_X, G_Y, x_X, x_Y) = -\frac{\lambda_{\text{VAE}}}{N} \sum_{m=1}^N (\log p(\tilde{l}_Y | \gamma_X, \delta_Y) + \log p(\hat{p}_X | \gamma_X) + \log p(\tilde{p}_Y | \delta_Y)), \quad (10)$$

where \hat{p}_X denotes the inpainting used to translate the original image x_X to domain Y and N is the total number of pixels. \tilde{p}_X is the inpainting produced when translating an already translated image \hat{x}_X that originated from Y back to that domain. Similarly, \hat{l}_X and \tilde{l}_X denote the respective labelmaps:

$$\log p(\hat{p}_X | \gamma_X) = \frac{\|(1 - \hat{l}_X)(\hat{p}_X - x_X)\|_2}{\omega_1}, \log p(\tilde{p}_X | \delta_Y) = \frac{\|\tilde{l}_Y(\tilde{p}_X - x_Y)\|_2}{\omega_2} \quad (11)$$

where $\omega_1 = \frac{(\sum(1 - \hat{l}_X) + \varepsilon)}{N}$ and $\omega_2 = \frac{(\sum(\tilde{l}_Y) + \varepsilon)}{N}$ are considered constant during optimization, with $\varepsilon > 0$. Finally, we use the labelmap produced by the other generator responsible for the opposite transformation \tilde{l}_Y as ground truth for \tilde{l}_X , where we consider \hat{l}_Y constant in this term:

$$\log p(\tilde{l}_X | \gamma_X, \delta_Y) = \hat{l}_Y \log \tilde{l}_X + (1 - \hat{l}_Y) \log(1 - \tilde{l}_X). \quad (12)$$

To restrict the solution space of our model, we use \mathcal{L}_{VAE} for both directions.

Identity Loss We apply an identity loss [10] on labelmap l_{X, x_Y} which results from feeding G_X with the wrong input x_Y . In this case G_X should not change anything, since the input is already in the desired domain Y :

$$\mathcal{L}_{\text{Idt}}(G_X, x_Y) = \lambda_{\text{Idt}} \|l_{X, x_Y}\|_1. \quad (13)$$

Relevancy Loss By now, we have defined all necessary constraints for a successful translation between image domains. The remaining constraints restrict the location and amount of change, l_X . Fulfilling Eq. (5), we want to entice label map l_X to be only set at locations of a large difference between inpainting p_X and image x_X and penalize large label maps in general:

$$\mathcal{L}_{\text{R}}(G_X, x_X) = \lambda_{\text{R}} \left[\left\| -\log(1 - l_X^2) \right\|_1 - \frac{\|l_X(x_X - p_X)\|_1}{\|l_X\|_1} \right]. \quad (14)$$

In order to not reward exaggerated pathology inpaintings, we consider $(x_X - p_X)$ constant in this expression.

Full PathoGAN Objective combining all loss terms for direction X to Y as $\mathcal{L}_{X \rightarrow Y}$, we can finally define:

$$\mathcal{L}_{X \rightarrow Y} = \mathcal{L}_{\text{GAN}} + \mathcal{L}_{\text{CC}} + \mathcal{L}_{\text{VAE}} + \mathcal{L}_{\text{Idt}} + \mathcal{L}_{\text{R}}, \quad (15)$$

$$\mathcal{L}_{\text{PathoGAN}} = \mathcal{L}_{\mathcal{A} \rightarrow \mathcal{B}} + \mathcal{L}_{\mathcal{B} \rightarrow \mathcal{A}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}(x_{\mathcal{A}}, x_{\mathcal{B}}). \quad (16)$$

2.4 Training

We include all training patients of Brats2017 and normalize each brain scan to follow $\mathcal{N}(0, 1/3)$, excluding zero voxels, and clip the resulting intensities to $[-1, 1]$. We select all slices from 60 to 100. In order to create two distinct datasets and relying on the manual segmentations, we label slices without pathology as *healthy*, with more than 20 pixels segmented as *pathological* slices, and discard the rest. For training, we select 1500 unaffected and 6000 pathological slices from a total of 1755 and 9413 respectively¹.

Since the BratS evaluation is volumetric and comparing performance is difficult, we also train a supervised segmentation technique on our data. We chose MDGRU [1] for this task, a multi-dimensional recurrent neural network, due to code availability and consistent state-of-the-art performance on different datasets.

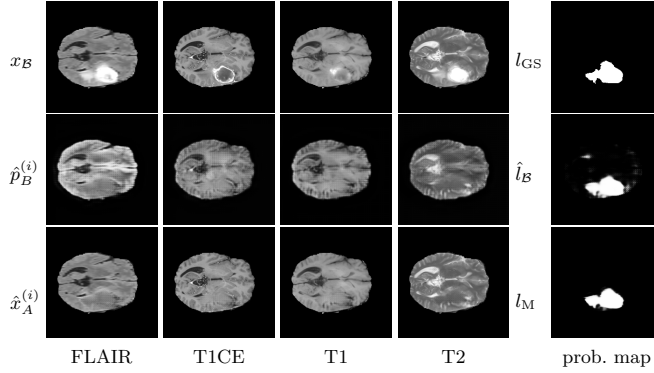


Fig. 2. Qualitative Results of one testing data example. *Left columns, top to bottom:* the four available image channels $x_B^{(i)}$, the generated inpaintings $\hat{p}_B^{(i)}$ and the translated images $\hat{x}_A^{(i)}$. *Right column, top to bottom:* The manual segmentation l_{GS} , the probability maps \hat{l}_B from PathoGAN and l_M from MDGRU for whole tumor.

¹ Thus we would like to stress that the manual segmentations were only used to create the two image domains, but not for the actual training.

3 Results

We train PathoGAN² for 119 epochs using batches of 4 and $\lambda_{KL} = 0.1, \lambda_R = 0.5, \lambda_{idt} = 1, \lambda_{CC} = 5$ and $\lambda_{VAE} = 1$. We trained MDGRU³ as defined in [1], using batches of 4 and 27 500 iterations. During training of both PathoGAN and MDGRU, we use weak data augmentation [1]. Table 1 shows the results on the pathological training and test data. On the left, Fig. 2 shows an exemplary sample from the testset, with generated inpaintings and translation result. On the right, we provide the generated labelmap together with the manual label for “whole tumor” and the computed segmentation with MDGRU. Details on the used architecture and training procedure as well as further qualitative samples are described in the supplementary material.

Table 1. Segmentation Results. *Columns:* Dice, 95th percentile Hausdorff distance (HD95), average Hausdorff distance (AVD) and volumetric Dice per-patient (Dice PP) by stacking all evaluated slices. *Rows:* Scores are shown as mean \pm std(median) for PathoGAN (proposed) and MDGRU, applied to training (Tr) and testing (Te) data.

	Dice (in %)	HD95 (in pixel)	AVD (in pixel)	Dice PP (in %)
PathoGAN (Tr)	72.4 \pm 24.4(81.0)	40.6 \pm 30.7(38.0)	10.3 \pm 15.4(4.7)	77.4 \pm 14.4(81.2)
PathoGAN (Te)	72.9 \pm 23.8(81.4)	39.4 \pm 29.9(37.6)	9.4 \pm 13.7(4.6)	77.4 \pm 14.4(81.7)
MDGRU (Tr)	87.8 \pm 20.0(94.4)	3.7 \pm 9.7(1.0)	1.0 \pm 4.7(0.2)	90.8 \pm 8.8(93.3)
MDGRU (Te)	86.3 \pm 21.3(93.6)	3.9 \pm 9.5(1.0)	1.1 \pm 4.9(0.2)	90.6 \pm 9.5(93.1)

4 Discussion

The results in Fig. 2 indicate that our relative weighting of the two inpainting reconstruction losses results in better reconstruction inside the tumor region than outside. The labelmaps of the supervised method compared to ours in Fig. 2 show great agreement, and both are relatively close to the gold standard. As the 95th-percentile and average Hausdorff measures in Table 1 show, there are some outliers in our proposed method, due to its weakly-supervised nature. The Dice score is about 10% smaller for both the per-slice as well as the per-patient case, given no labels are provided. It is important to remember that we segment with the only criterion of being not part of the healthy distribution, which could vary from the subjective measures used to manually segment data. The increase in accuracy and decrease in standard deviation in the per-patient case for both

² Our implementation is based on <https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix>.

³ We use the implementation of MDGRU at <https://github.com/zubata88/mdgru>.

methods is most likely caused by the inferior segmentation performance in slices showing little pathology. The per-patient Dice of the supervised method is in the range of the top methods of BraTS 2017. Although not directly comparable, this suggests that we can use our computed supervised scores as good reference to compare our results to.

We did only scratch the surface on the possible applications of our proposed formulation. Future work will include unaffected samples that are actually healthy. Furthermore, the model architecture could be drastically simplified using one discriminator for both directions, allowing for larger generator networks as well as using multiple discriminators at different scales to find inpaintings that are not just locally but also globally consistent with the image. A restriction to slices is unfortunate but necessary due to memory requirements. A generalisation of our approach to volumetric data would make it feasible for more real clinical applications.

Conclusion We presented a new generative pathology segmentation model capable of handling a plethora of tasks: First and foremost, we presented a weakly supervised segmentation method for pathologies in 2D medical images, where it is only known if the image is affected by the pathology. Furthermore, we were able to sample from both our healthy as well as our pathology model. We showed qualitatively and quantitatively, that we are able to produce compelling results, motivating further research towards actual clinical applications of PathoGAN.

References

1. Andermatt, S., Pezold, S., Cattin, P.: Multi-dimensional Gated Recurrent Units for the Segmentation of Biomedical 3D-Data. In: International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis. pp. 142–151. Springer (2016)
2. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J., Farahani, K., Davatzikos, C.: Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features. *Nature Scientific Data* (2017), [in press]
3. Baumgartner, C.F., Koch, L.M., Tezcan, K.C., Ang, J.X., Konukoglu, E.: Visual feature attribution using wasserstein gans. *arXiv preprint arXiv:1711.08998* (2017)
4. Chu, C., Zhmoginov, A., Sandler, M.: CycleGAN, a Master of Steganography. *arXiv:1712.02950 [cs, stat]* (Dec 2017)
5. Fu, C., Lee, S., Ho, D.J., Han, S., Salama, P., Dunn, K.W., Delp, E.J.: Fluorescence microscopy image segmentation using convolutional neural network with generative adversarial networks. *arXiv preprint arXiv:1801.07198* (2018)
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: *Advances in neural information processing systems*. pp. 2672–2680 (2014)
7. Kingma, D.P., Welling, M.: Stochastic gradient vb and the variational auto-encoder. In: *Second International Conference on Learning Representations* (2014)
8. Menze, B.H., Jakab, A., Bauer, S., et al: The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging* 34(10), 1993–2024 (Oct 2015)

9. Tsaftaris, S.A.: Adversarial image synthesis for unpaired multi-modal cardiac data. In: Simulation and Synthesis in Medical Imaging: Second International Workshop, SASHIMI 2017, Held in Conjunction with MICCAI 2017. p. 3. Springer (2017)
10. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. pp. 2223–2232 (2017)

Chapter 9

Automated Tracking of Lesions

Introduction

This work covers the post-processing of lesion maps, a statistical analysis of individual lesion development in longitudinal studies. This is accomplished by means of coregistration of the lesion maps using the originating MR scans and cooccurrence information of individual lesions in the different maps. Ambiguity in lesion connections such as occurrence of confluent lesions in follow up scans is dealt with using the Bron-Kerbosch algorithm to find the maximum cliques, corresponding to the individual lesion developments. The paper has been coauthored with Athina Papadopoulou, where the workload was divided such, that the implementation and evaluation of the method as well as the description of the followed methodology in the paper was contributed by Simon Andermatt, whereas the medical details and the validation study were written by Athina Papadopoulou.

Publication This work has been published in the Journal of Neuroimaging as *Tracking the Evolution of Cerebral Gadolinium-Enhancing Lesions to Persistent T1 Black Holes in Multiple Sclerosis: Validation of a Semiautomated Pipeline*¹.

¹The publication is accessible at <https://doi.org/10.1111/jon.12439>

Tracking the Evolution of Cerebral Gadolinium-Enhancing Lesions to Persistent T1 Black Holes in Multiple Sclerosis: Validation of a Semiautomated Pipeline

Simon Andermatt*, Athina Papadopoulou*, Ernst-Wilhelm Radue, Till Sprenger, Philippe Cattin

From the Department of Biomedical Engineering, University of Basel, Basel, CH (SA, PC); Neurology Clinic and Policlinic, University Hospital Basel and University of Basel, Basel, CH (AP, TS); Medical Image Analysis Center, Basel, CH (AP, EWR, TS); and Department of Neurology, DKD HELIOS Klinik Wiesbaden, Wiesbaden, Germany (TS).

ABSTRACT

BACKGROUND: Some gadolinium-enhancing multiple sclerosis (MS) lesions remain T1-hypointense over months (“persistent black holes, BHs”) and represent areas of pronounced tissue loss. A reduced conversion of enhancing lesions to persistent BHs could suggest a favorable effect of a medication on tissue repair. However, the individual tracking of enhancing lesions can be very time-consuming in large clinical trials.

PURPOSE: We created a semiautomated workflow for tracking the evolution of individual MS lesions, to calculate the proportion of enhancing lesions becoming persistent BHs at follow-up.

METHODS: Our workflow automatically coregisters, compares, and detects overlaps between lesion masks at different time points. We tested the algorithm in a data set of Magnetic Resonance images (1.5 and 3T; spin-echo T1-sequences) from a phase 3 clinical trial ($n = 1,272$), in which all enhancing lesions and all BHs had been previously segmented at baseline and year 2. The algorithm analyzed the segmentation masks in a longitudinal fashion to determine which enhancing lesions at baseline turned into BHs at year 2. Images of 50 patients (192 enhancing lesions) were also reviewed by an experienced MRI rater, blinded to the algorithm results.

RESULTS: In this MRI data set, there were no cases that could not be processed by the algorithm. At year 2, 417 lesions were classified as persistent BHs ($417/1,613 = 25.9\%$). The agreement between the rater and the algorithm was $> 98\%$.

CONCLUSIONS: Due to the semiautomated procedure, this algorithm can be of great value in the analysis of large clinical trials, when a rater-based analysis would be time-consuming.

Keywords: Automatic tracking, MRI, enhancing lesions, hypointense lesions, algorithm.

Acceptance: Received December 19, 2016. Accepted for publication February 27, 2017.

Correspondence: Address correspondence to Athina Papadopoulou, Neurology Clinic and Policlinic, University Hospital Basel and University of Basel, Petersgraben 4, Basel, CH-4031, Switzerland. E-mail: athina.papadopoulou@usb.ch.

*These authors contributed equally to the paper.

Funding: This study was supported by a grant from Novartis Pharmaceuticals Switzerland.

Financial interests of the authors: SA has nothing to disclose. AP has received travel support from Bayer Health Care Pharmaceutical, Teva, and UCB Pharma, as well as grants from the University of Basel and the Swiss MS society. The institution of AP received payments for speaking from Genzyme and Actelion. EWR has received payment for lectures by Genzyme and Novartis. He is a consultant for the University Hospital Basel (Neurological and Psychiatric Clinics), for MIAC (Medical Image Analysis Center), c/o, and for Springer Verlag Publisher in Heidelberg (Germany). TS: The current (DKD Helios Klinik Wiesbaden) or previous (University Hospital Basel) institutions of TS have received payments for speaking or consulting from: Biogen Idec, Eli Lilly, Allergan, Actelion, ATI, Mitsubishi Pharma, Novartis, Genzyme, and Teva. Dr. Sprenger received research grants from the Swiss MS Society, Novartis Pharmaceuticals Switzerland, EFIC-Grünenthal grant, and Swiss National Science foundation. PC has nothing to disclose.

J Neuroimaging 2017;27:469-475.
DOI: 10.1111/jon.12439

Introduction

Magnetic resonance imaging (MRI) is currently the most important paraclinical tool to measure disease-related damage in the central nervous system of patients with multiple sclerosis (MS). The effect of an immunomodulatory treatment on the reduction of new MRI lesions over time seems to correlate well with the treatment effect on relapses¹⁻³ and disability accumulation. Thus, in clinical trials, MRI lesions are important surrogate markers of disease activity and treatment effects.

“Black holes” (BHs) are a specific subgroup of MS lesions that are hypointense on T1-weighted sequences. Although most gadolinium (Gd)-enhancing lesions are acutely T1-hypointense, only a subgroup remains hypointense for 6 months and longer after their first appearance.⁴ Such persistently T1-hypointense lesions are often called “persistent black holes.”⁵ These lesions

represent more pronounced, irreversible tissue damage and less remyelination as compared to T2 hyperintense, but T1 isointense chronic lesions.⁶⁻⁸ Thus, a reduction in the formation of persistent BHs through a medication would suggest a treatment effect on more severe tissue damage and/or remyelination.⁹ Accordingly, persistent BHs have been shown to correlate better to clinical disability accrual than T2 hyperintense lesions.¹⁰

A change in the volume of BHs throughout a clinical study is a typical secondary end point used to indicate a potential medication effect on the formation of new BHs in MS trials. However, this measure also relates to the treatment effect on the lesion formation in general (ie, patients with fewer newly appearing lesions under a given medication will probably also have fewer persistent BHs). Thus, the *proportion* of new Gd-enhancing lesions becoming persistent BHs over time is a

preferable, more specific measure to reflect a treatment effect on repair mechanisms and remyelination. To assess this measure, tracking of all Gd-enhancing lesions over time is needed. However, such a manual/visual tracking of individual lesions can be very time-consuming and hence expensive. Thus, we aimed at creating a method that can track the evolution of individual MS lesions longitudinally in a semiautomated way, on the basis of previously segmented lesion maps.

Our objective was to create a quick, automated method to easily calculate the proportion of Gd-enhancing lesions that become persistent BHs at follow-up, based on lesion maps, for a use mostly in clinical trials.

Materials and Methods

Study Population and Data Acquisition

MRI data from the international multicenter, randomized-controlled, phase 3 FREEDOMS¹¹ trial comparing fingolimod with placebo in MS were used in this work. Details on the study population and MRI acquisition parameters have been previously published.¹¹ In brief, this was a double-blind, randomized study that enrolled patients with relapsing-remitting MS. The study had three arms: .5 mg fingolimod, 1.25 mg fingolimod, and placebo. Brain MRI was performed using a standardized protocol at baseline, 6, 12, and 24 months. The MRI protocol included a whole-brain T1-weighted spin-echo sequence (3 mm slice thickness), before and after injection of Gd. Both 1.5 and 3 Tesla magnets were allowed. All MRIs at baseline ($n = 1,272$ patients) were included in this analysis.

In this investigator-initiated methodological MRI study, we were fully blinded to the treatment assignment of the patients, and thus unable to assess a possible treatment effect.

Preprocessing/Lesion Segmentation

For the algorithm to work, all lesions had to be previously segmented, to obtain masks that could be subjected to the longitudinal tracking analysis. The lesion segmentations had previously been performed as part of the original analysis of the clinical trial¹¹ in a semiautomated way, using "AMIRA" (Mercury Computer Systems Inc., Andover, Massachusetts, USA). No intensity normalization was used during the process of lesion segmentation, but a manual adjustment of the image intensity (and contrast) of the images by experienced raters was possible, when needed.

At baseline and at year 2, the following lesion types were available as lesion segmentation maps in the used data set: (1) Gd-enhancing lesions on postcontrast T1-weighted images, (2) hyperintense lesions on proton density (PD)/T2-weighted images, and (3) BHs on postcontrast T1-weighted spin-echo images. BHs were assessed on postcontrast images in order to exclude acute BHs (concurrent with contrast enhancement). BHs were defined as lesions with a signal intensity that was lower than the surrounding normal appearing white matter and at least as low as the normal appearing cortical gray matter. A BH was only considered if it corresponded to a PD/T2 hyperintense lesion.

The mean T2-lesion volume of all patients at baseline was 6,368 mm³ (SD 7,755).

The segmentation masks were registered per patient. We used the baseline PD-weighted scans as reference to register all other volumes of the same patient. All registrations were

performed using the rigid registration implemented in the niftyreg suite.^{12,13} Registration quality was ensured by visual inspection. There were no cases that had to be excluded due to poor registration. The lesion masks were registered to the PD reference using the transformations calculated from the registrations of their corresponding scans (see A and B in Fig 1).

Algorithm

For each patient, every lesion in the available lesion maps was identified. For the baseline and the follow-up, each lesion's type was determined according to the MRI sequences/segmentation masks it was found in. For example, if one lesion was found as part of the Gd-enhancing mask and the PD-mask at baseline, it was defined as a "Gd-enhancing lesion"; if a lesion was found as part of the BH-mask and the PD-mask at follow-up and not of the Gd-enhancing mask at follow-up, it was defined as "black hole." For each follow-up, the lesions were matched to the corresponding lesion in earlier scans (if any), and the changes in type were registered (eg, from "Gd-enhancing lesion" at baseline to "black hole" at follow-up). Hereby, we considered lesions to be of the same origin across scans, if at least one full voxel of said lesions in the respective lesion maps was overlapping. To accomplish this task, we used the following technique.

To detect each individual lesion, the lesion masks were analyzed through connected component labeling, which labels each voxel in a group of connected voxels with the same number (Fig 1D). To facilitate the process of relating lesions to one another computationally, a superimposition per patient was created by adding all masks together (Fig 1C). This step divides the problem of finding lesion connections into smaller subproblems, which can be solved more efficiently. We applied connected component labeling to the superimposition mask as well, considering every value larger than zero and received clusters, which corresponded to lesions that might be related to one another. In a first step, every voxel cluster of each mask was compared to all the clusters of the superimposition mask. If a voxel correspondence was found, that lesion was grouped into the lesion group detected in the superimposition mask (Fig 1E). The amount of overlapping voxels did not matter in this step, since we know that if there is an overlap, the whole lesion must be enclosed in the group.

In the next step, each lesion was compared with all other lesions of the same lesion group (F) using additive correspondence. Additive correspondence was assumed, if the sum of the products of all overlapping voxels was above a given threshold τ . To better understand why this is a meaningful property, we can look at the interpolated values as the probability of the respective voxel being set or not. The product of the interpolated voxel v_1 and v_2 represents the expectation value of overlapping area, if no prior information, eg the neighborhood of the voxel, is considered.

$$\begin{aligned} E[v_1 v_2 = 1] &= \sum_{a, b \in \{0, 1\}} p(v_1 = a) p(v_2 = b) v_1 v_2 \\ &= p(v_1 = 1) p(v_2 = 1) \end{aligned}$$

Due to inconclusive lesion borders, difference in expert opinion, registration errors, and small or numerous lesions, it is often the case that there are some groups where more than one lesion of the same mask and time point are grouped together. Since these are usually individual lesions, we split them

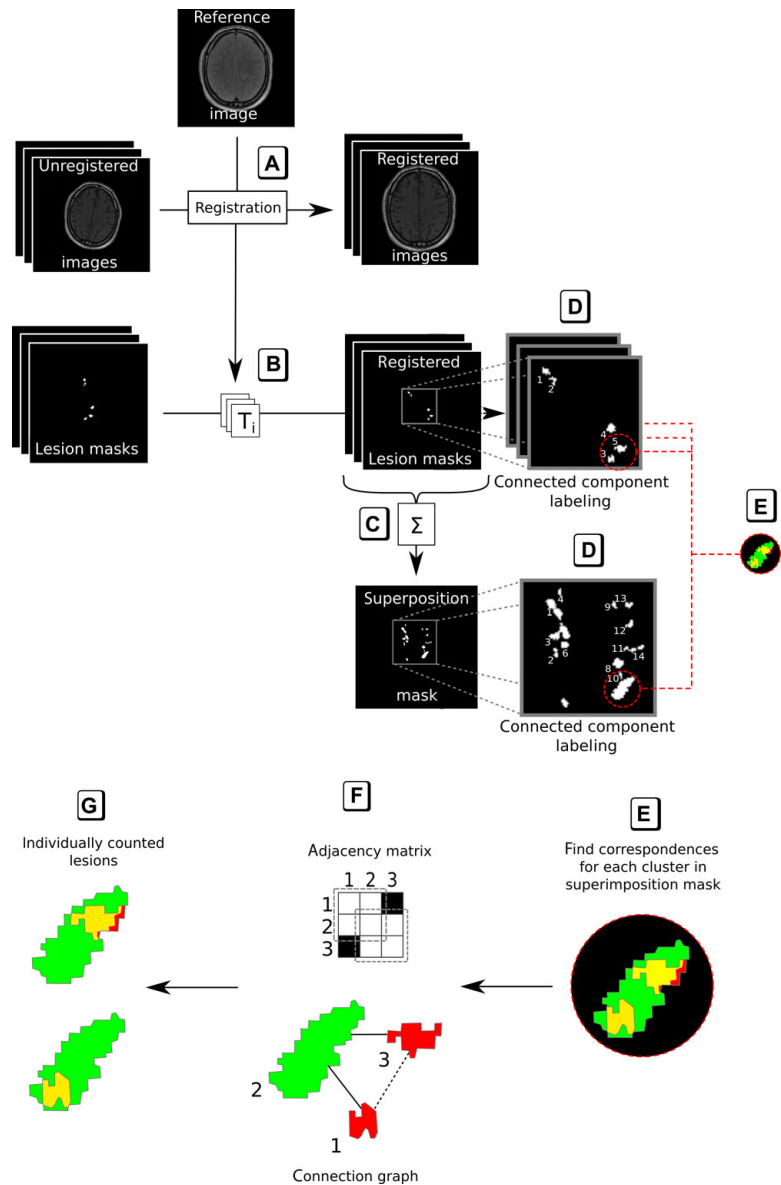


Fig 1. Workflow diagram. Diagram of the applied workflow including the necessary preprocessing steps. Given a data set where lesion masks were created on unregistered scans, all scans per patient have to be registered to the same space (A). The resulting transformations can be used to transform the lesion masks into the same reference system (B). All masks per patient are then merged together to create a superimposition mask, where each voxel is set to one at whose position a lesion was found in at least one of the masks (C). Connected component analysis is applied to all masks as well as to the superimposition mask to identify individual lesions (D). For each cluster that was detected in the superimposition mask, we apply steps (E)-(G). First, we try to find all lesions in all masks that contributed to the cluster (E). For each of these lesions, we identify all other lesions in the same cluster that have an overlap that is at least of threshold τ (in this case, one voxel) (F). In the connection graph, overlaps are visualized with a solid line and missing connections with a dashed line. The same graph can be visualized with an adjacency matrix with values set to 1 where a connection exists and 0, where no overlap was present (visualized with white and black squares). Finally, we use the adjacency matrix to identify maximum cliques corresponding to individual lesions (G). For details, see also section "Algorithm."

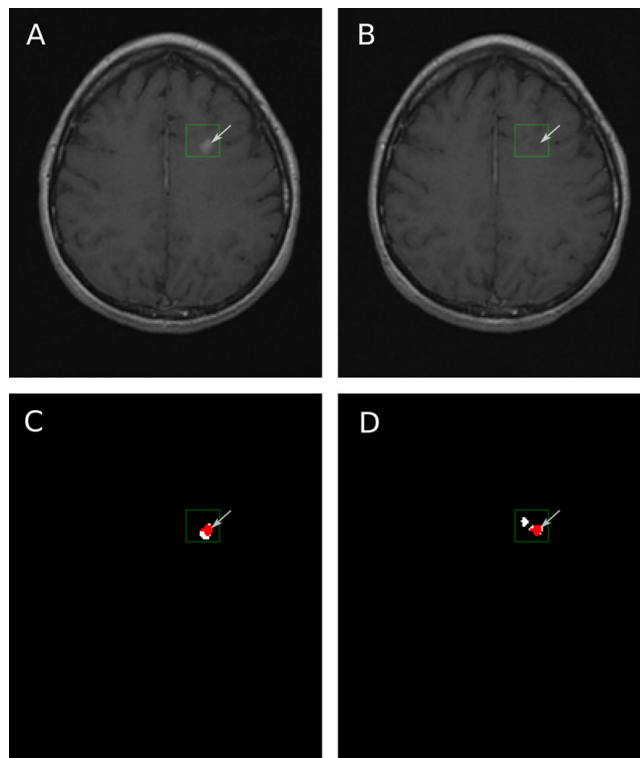


Fig 2. Example of a lesion tracking from baseline to follow-up. Evolution of a gadolinium (Gd)-enhancing lesion at baseline to a T1-black hole at follow-up and evaluation through the algorithm. The red area depicts the overlapping area between the two lesion masks (Gd-lesion mask at baseline and black hole (BH)-mask at follow-up), indicating that the Gd-enhancing lesion at baseline was classified by the algorithm as persistent black hole at year 2. (A): Gd-enhancing lesion at baseline on T1-weighted spin-echo sequence after contrast agent (Gd) administration. (B): The lesion is T1-hypointense at year 2 ("persistent black hole"). (C): Segmentation mask of Gd-enhancing lesions at baseline. (D): Segmentation mask of T1-black holes at year 2. (C) and (D): In white, one can see the entire lesion masks and in red the overlapping pixels between the two masks. Since there was an overlap of more than one voxel between the Gd-mask at baseline and the BH-mask at year 2, this lesion was classified as "persistent black hole."

into individual subgroups, counting one or more large lesions multiple times due to the fact that they enclose more than one smaller lesion.

Figures 1E-G show an example of the process of splitting up a lesion cluster consisting of a confluent-enhancing lesion at baseline, which forms two separate persistent BHs. To determine how many individual lesions could be found in one cluster, we used the notion of maximal clique. Each lesion in the superimposition mask is made up of n lesions originating from one or more different masks (color-coded in orange, green, and yellow in the example). By comparing each lesion with all other lesions of the same cluster, we receive a connection graph (F). This graph can be represented with an n times n adjacency matrix, which represents the connections between all lesions in one cluster. We denote a connection with one, and zero otherwise. Finding the largest fully connected subgraphs in the adjacency matrix results in the desired separated lesion groups (G). This task is identical to the determination of all maximal cliques. A clique is a fully connected graph, in our case a group of lesions where each lesion features an overlap with every other lesion in the group. A maximal clique is a clique, where

there is no other lesion outside the clique, which overlaps every lesion in the clique. These can be determined with the Bron-Kerbosch algorithm.¹⁴ Each maximal clique in every group is hence counted as one lesion.

Analysis

In order to assess the proportion of Gd-enhancing lesions at baseline that developed into persistent BHs at year 2, the following analysis was performed: The segmentation masks of Gd-enhancing lesions at baseline were compared with the segmentation masks of BHs at year 2. If there was an overlap of at least one voxel, the corresponding lesion to which this voxel belonged to was automatically classified as "persistent black hole." An example of this process is shown in Figure 2.

Comparison with Rater Analysis

To confirm the capability of the method to compare the masks and link the correct lesions at the different time points, we designed a validation test. We randomly selected images of 50 patients from the study sample to be reviewed by an experienced MRI rater (AP), blinded to the results of the algorithm.

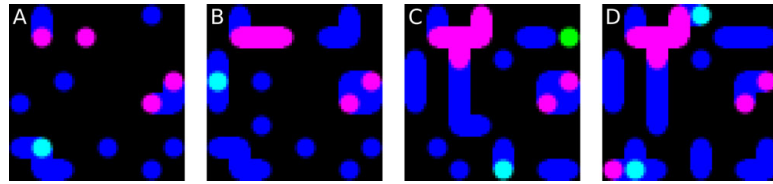


Fig 3. Example of the simulation process for the validation of the algorithm. Simulation results of time points 4, 6, 8, and 10, given a grid of 8 times 8 potential artificial lesions. The different stages are color coded and superimposed. Each lesion mask is represented by one color: red for black holes (BHs), green for gadolinium (Gd)-enhancing lesions, and blue for PD/T2-lesions. Pink is the combination of red and blue and indicates that this lesion area corresponds to the red (BH) mask and the blue (PD/T2) mask and thus represents a T2-lesion with a BH. Similarly, light blue corresponds to the combination of blue and green and thus represents a T2-lesion which is at the same time Gd enhancing. The probabilities for a transition of one state to the other are given in Table 1.

Table 1. Transition Probabilities Used for the Simulation from Time Point tp_i to Time Point tp_{i+1}

		Time Point tp_{i+1}							
		Nothing	Gd	BH	GdBH	PD	PDGd	PDBH	PDGdBH
Time point tp_i	Nothing	.98	.02	0	0	0	0	0	0
	Gd	.1	.1	0	0	0	.8	0	0
	BH	0	0	1	0	0	0	0	0
	GdBH	0	0	0	1	0	0	0	0
	PD	.01	0	0	0	.99	0	0	0
	PDGd	0	0	0	0	.6	.1	.3	0
	PDBH	0	0	0	0	0	0	1	0
	PDGdBH	0	0	0	0	0	0	0	1

BH = black holes–hypointense lesions on the T1-weighted images postcontrast; Gd = gadolinium (Gd)-enhancing lesions on the T1-weighted images postcontrast; PD = hyperintense lesions on PD/T2-weighted images. Any combination of the before mentioned types describes a lesion occurring in the respective masks (ie, a PDGd lesion represents a T2-lesion with Gd-enhancement).

The rater visually compared the same masks at baseline and year 2 and noted if there was an overlap (persistent BH) or not.

Simulated Data

In addition, we created a simulation to show the performance of the algorithm in applications, where multiple types of lesion masks and more than two time points are taken into consideration. The simulation was performed by creating a grid of size $n \times m$, where each point on the grid corresponds to a potential lesion. Each point is characterized by a combination of appearances in different scans, defining its lesion type. Each grid point was initialized as not appearing in any lesion map. With specific probabilities for transitions to a new type for any given lesion type, artificial follow-up scans were created. The resulting time series was then visualized on an image of a greater resolution, where lesions of the same type that were adjacent to each other were merged together. Figure 3 shows an example of this process and Table 1 shows the probabilities that were used for the transition between types.

Results

Percentage of Gd-Enhancing Lesions Converting to Persistent Black Holes

A total of 1,613 Gd-enhancing lesions were tracked at baseline in a total of 400 patients. The rest of the patients had no Gd-enhancing lesions at baseline or did not receive a follow-up MRI at year 2. From these 1,613 lesions, 417 (25.9%) were classified as persistent BHs at year 2. The rest were considered to have become T1-isointense. All lesions were evaluated by

the algorithm (including images acquired at 1.5 and 3T MRI field strength).

There were no cases that could not be processed by the algorithm. Moreover, the algorithm was very quick; the processing of these 400 patients (MRIs at two time points: baseline and year 2, with previously completed lesion segmentation and registration) was performed in approximately 2 minutes on a standard workstation. If the maps had to be registered prior to the analysis using the sequences they originated from, the rigid registration per volume would be around 2 minutes each.

Validation of the Algorithm through Comparison with Rater

In the subgroup of 50 random patients (192 lesions), the rater (AP) and the algorithm were in agreement in the vast majority of lesions (189/192 lesions = 98.4%). The three lesions with inconsistent results were all rated by the rater as T1-isointense (not persistent BHs) and by the algorithm as persistent BHs. When the three lesions were reevaluated by two raters (AP and SA), they were all reclassified to be indeed persistent BHs, since small overlaps of at least one voxel between the BH-mask at year 2 and the Gd-enhancing-mask at baseline could be detected.

The lesion load of these 50 randomly chosen patients was relatively high (mean T2-lesion volume at baseline: $9,464 \pm 9,650 \text{ mm}^3$).

Validation of the Algorithm with Simulated Data

In Figure 3, simulated data are shown for time points 4, 6, 8, and 10 of a total of 10 time steps on a grid of 8 times 8. The lesions in the simulated masks for T1-weighted MRI, T1-weighted MRI

with contrast agent, and PD-weighted MRI are color coded in red, green, and blue, respectively. Occurrences in multiple masks are shown as mixed color (e.g. pink is the combination of red and blue).

A large number of lesions were modeled to mimic a severely affected area, where some lesions merge into larger lesions to make the application of the maximal clique splitting technique necessary. The data shown in Figure 3 were evaluated automatically using our method, as well as manually by counting the number of lesions that formed individually over time. Prior knowledge about the simulation or form of lesions was not used for the manual evaluation.

Both manual and automated evaluations returned 21 individual lesion developments of which four were related to our task of finding persistent BHs. All four transitions resulted in enhancing lesions not forming a persistent BH for both the manual and the automated evaluation (correct performance of the algorithm for all four transitions). Figure 4 shows the indices of the 21 individual lesions that were found by the algorithm.

Discussion

In this study, we showed that an automated method can easily compare segmented lesion masks at different time points, to assess the percentage of enhancing lesions that form persistent BHs over time. The algorithm worked very well and extremely quickly (few minutes) in this relatively large data set and showed an excellent agreement with the assessment of an experienced rater.

The most important advantage of our method is its automated postprocessing character, which offers objective and quick results, compared to a rater-based analysis. Thus, we believe that this method would be ideal for analysis of large data from clinical trials, with MR images of many patients at different time points and already available lesion segmentation masks.

Regarding the influence of image-related parameters on our method, it needs to be stressed that these affect only the pre-processing of the data (ie, the lesion segmentation). The algorithm itself does not work with raw data, but instead uses lesion masks, therefore, we believe that the algorithm would perform similarly well with data derived from T1 spin echo as well as with data from T1 gradient echo sequences. However, only T1 spin-echo data were available in the current study and we have not systematically tested the performance based on T1 gradient

echo data, which have been shown to be more sensitive for the detection of T1 hypointense lesions in MS.¹⁵

To summarize, our method presented here cannot replace the detection (marking) and segmentation of lesions, since the masks have to be available for the algorithm to work. However, it can dramatically decrease the time needed to assess the proportion of enhancing lesions becoming persistent BHs. Since this proportion can be used as a surrogate of remyelination and tissue repair, the algorithm would be of particular interest in clinical studies testing medications with potential effects on remyelination and tissue repair.

Our results, with a 25.9% proportion of Gd-enhancing lesions converting to persistent BHs at 2 years of follow-up, are compatible with the literature (variable results in untreated patients ranging from approximately 20% to 40%,^{16–23} depending on the selection and number of patients, the MRI parameters, and the duration of the follow-up). The fact that our results are on the relatively lower end of those described in the literature is not surprising, since we included patients on placebo and patients on fingolimod which probably reduced the risk of BH formation.^{11,17}

Apart from the use in large clinical trials, the algorithm has the potential to be used in more explorative MRI studies. Similarly to the comparison of Gd-enhancing- and BH-masks, other lesion transitions can be tested. For example, individual lesions that are initially visible on PD-weighted images, but disappear at follow-up (probably due to very efficient repair) could be easily detected by the algorithm, by comparing the PD-masks at baseline with the PD-masks at follow-up. Studying such subgroups of lesions and comparing the frequency of “favorable” lesion evolutions between treatment groups in clinical trials could be particularly interesting and could be easily performed by our algorithm—if segmentation masks are available. Our evaluation of simulated data indicates that the algorithm can be also applied in more complicated scenarios, eg, with more than one lesion map per screening and more than two points.

As mentioned above, image-related parameters are not expected to influence the performance of the algorithm. However, there is one parameter that does: the minimum overlap of lesions to define a connection between follow-up and baseline. In this paper, we chose a minimum overlap of one voxel for this purpose. This could be regarded as too low and prone to errors in registration. Depending on the application and the MRI data set (eg, image resolution, cutoff of the lesion size for segmentation, etc), a higher cutoff could be more appropriate. For this

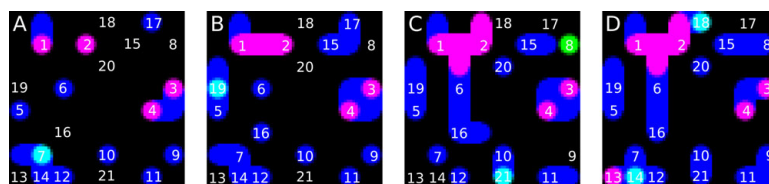


Fig 4. Results of the analysis of the lesion maps A, B, C and D shown in Figure 3. The indices in the picture refer to individual lesions detected by the algorithm. The indices are not necessarily centered, but are located at the position of one of the associated grid points. The clusters in the superimposition mask that were split into the given indices consisted of the following lesions: (1,2,6,16,18), (3,4), (5,19), (7,12,13,14), (8,15,17), (9,11), (10,21), and (20). As an example of the splitting process, the following explains how the first cluster was split into the given indices. Since 1, 2, and 6 were separated from one another in the first image (A), they were counted as individual lesions. Lesion 16 appeared as a new individual lesion in image (B) and 18 appeared as a new gadolinium-enhancing lesion in image (D) (light blue). Both 16 and 18 were counted correctly as additional individual lesions, even though all lesions (1, 2, 6, 16, and 18) are joined in image (D).

reason, the threshold applied to determine if lesions are of the same origin is denoted in the algorithm as “variable τ ” and can be changed if needed.

There are of course some limitations of our method.

An important limitation relates to the difficulty of the algorithm to distinguish between confluent lesions that had been segmented as one larger area at baseline. Indeed, the simulation showed that in theoretical cases with many confluent T2-lesions at baseline, the algorithm would count the confluent lesions as one and would not assess their evolution separately (despite the use of the “lesion splitting technique” shown in Fig 4). Thus, the connections between follow-up and baseline were underestimated in such cases. A similar problem would rise if there would be many new lesions appearing in between baseline and follow-up scan, near to/confluent with baseline lesions. However, this main drawback of the algorithm should not be a problem for the specific application on determining the evolution of Gd-enhancing lesions to BHs, since multiple, confluent Gd-enhancing lesions at baseline represent an infrequent scenario. Moreover, the simulation showed that in all scenarios where the evolution of “Gd lesion converting to BH” was assessed, the performance of the algorithm was correct.

Another possible limitation of the workflow lies in the fact that we assume perfect alignment of the different brain scans. Due to artifacts in MR scans, unrelated pathologies, inconsistent segmentation, and/or inaccurate registration, some lesions of small size might not overlap between scans. In such a case, instead of one connection, two disconnected lesions would be counted. To omit this behavior, a more sophisticated registration method such as free-form registration²⁴ could be applied, although some preprocessing of the data might be necessary (prior lesion removal and/or specific constraints).

In conclusion, we here developed a semiautomated workflow for tracking the evolution of enhancing MS lesions longitudinally. Our method can be used in clinical trials with large MRI data sets and available lesion masks, to detect potential treatment effects on tissue repair.

References

1. Sormani MP, Bruzzi P. MRI lesions as a surrogate for relapses in multiple sclerosis: a meta-analysis of randomised trials. *Lancet Neurol* 2013;12:669-76.
2. Sormani MP, Bonzano L, Roccatagliata L, et al. Magnetic resonance imaging as surrogate for clinical endpoints in multiple sclerosis: data on novel oral drugs. *Mult Scler* 2011;17:630-3.
3. Sormani MP, Bonzano L, Roccatagliata L, et al. Surrogate endpoints for EDSS worsening in multiple sclerosis. A meta-analytic approach. *Neurology* 2010;75:302-9.
4. Rovira A, León A. MR in the diagnosis and monitoring of multiple sclerosis: an overview. *Eur J Radiol* 2008;67:409-14.
5. Sahrana MA, Radue E-W, Haller S, et al. Black holes in multiple sclerosis: definition, evolution, and clinical correlations. *Acta Neurol Scand* 2009;122:1-8.
6. van Walderveen MA, Kamphorst W, Scheltens P, et al. Histopathologic correlate of hypointense lesions on T1-weighted spin-echo MRI in multiple sclerosis. *Neurology* 1998;50:1282-8.
7. Zivadinov R, Leist TP. Clinical-magnetic resonance imaging correlations in multiple sclerosis. *J Neuroimaging* 2005;15:10S-21S.
8. Zivadinov R, Bakshi R. Role of MRI in multiple sclerosis I: inflammation and lesions. *Front Biosci* 2004;9:665-83.
9. Oommen VV, Tauhid S, Healy BC, et al. The effect of fingolimod on conversion of acute gadolinium-enhancing lesions to chronic T1 hypointensities in multiple sclerosis. *J Neuroimaging* 2016;26:184-7.
10. Truyen L, van Waesberghe JH, van Walderveen MA, et al. Accumulation of hypointense lesions ('black holes') on T1 spin-echo MRI correlates with disease progression in multiple sclerosis. *Neurology* 1996;47:1469-76.
11. Kappos L, Radue E-W, O'Connor P, et al. A placebo-controlled trial of oral fingolimod in relapsing multiple sclerosis. *N Engl J Med* 2010;362:387-401.
12. Ourselin S, Roche A, Subsol G, et al. Reconstructing a 3D structure from serial histological sections. *Image Vis Comput* 2001;19:25-31.
13. Ourselin S, Stefanescu R, Pennec X. Robust registration of multimodal images: towards real-time clinical applications. *Med Image Comput Comput Assist Interv* 2002;5:140-7.
14. Bron C, Kerbosch J. Algorithm 457: finding all cliques of an undirected graph. *Commun ACM* 1973;16:575-7.
15. Dupuy SL, Tauhid S, Kim G, et al. MRI detection of hypointense brain lesions in patients with multiple sclerosis: T1 spin-echo vs. gradient-echo. *Eur J Radiol* 2015;84:1564-8.
16. Barkhof F, Hulst HE, Druilovic J, et al. Ibudilast in relapsing-remitting multiple sclerosis: a neuroprotectant? *Neurology* 2010;74:1033-40.
17. Oommen VV, Tauhid S, Healy BC, et al. The effect of fingolimod on conversion of acute gadolinium-enhancing lesions to chronic T1 hypointensities in multiple sclerosis. *J Neuroimaging* 2016;26:184-7.
18. van Waesberghe JH, van Walderveen MA, Castelijns JA, et al. Patterns of lesion development in multiple sclerosis: longitudinal observations with T1-weighted spin-echo and magnetization transfer MR. *Am J Neuroradiol* 1998;19:675-83.
19. Ciccirelli O, Giugni E, Paolillo A, et al. Magnetic resonance outcome of new enhancing lesions in patients with relapsing-remitting multiple sclerosis. *Eur J Neurol* 1999;6:455-9.
20. Filippi M, Rovaris M, Rocca MA, et al. European/Canadian glatiramer acetate study group. Glatiramer acetate reduces the proportion of new MS lesions evolving into “black holes”. *Neurology* 2001;57:731-3.
21. Dalton CM, Miszkil KA, Barker GJ, et al. Effect of natalizumab on conversion of gadolinium enhancing lesions to T1 hypointense lesions in relapsing multiple sclerosis. *J Neurol* 2004;251:407-13.
22. Bagnato F, Jeffries N, Richert ND, et al. Evolution of T1 black holes in patients with multiple sclerosis images monthly for 4 years. *Brain* 2003;126:1782-9.
23. van den Elskamp IJ, Lembecke J, Dattola V, et al. Persistent T1 hypointensity as an MRI marker for treatment efficacy in multiple sclerosis. *Mult Scler* 2008;14:764-9.
24. Modat M, Ridgway GR, Taylor ZA, et al. Fast free-form deformation using graphics processing units. *Comput Methods Programs Biomed* 2010;98:278-84.

Chapter 10

Discussion and Conclusion

Discussion

In this thesis, we have detailed a full set of tools to automatically segment MR images containing brain pathology. We proposed a new approach based on an efficient gated recurrent architecture for supervised segmentation. We introduced a segmentation approach in the weakly-supervised setting, using nothing more than an image-wide label if pathology is present or not. Finally, we presented a computationally simple post-processing method for the automated tracking of lesions in longitudinal studies.

We first introduced a new neural network for image segmentation which we denoted MD-GRU, an adaptation of GRU until then only used for sequential data. We accomplished this by interpreting the volumetric data as sequence of images, twice in each dimension, once in forward and once in backward direction. We showed on a number of public benchmarks that we were able to match and sometimes surpass the state of the art with our formulation. MD-GRU has since been adopted at the University Hospital as well as our sponsors at MIAC AG. Using the publicly available code for this project¹, anybody has the means to train a segmentation network, given labeled data for the specific task is available.

Creating a sufficiently large training corpus for supervised segmentation is a labor- and time-intensive task. We hence tried to find a method that does not require such a large amount of labeled information. Using only one binary image-level label stating the occurrence of pathology in the image, we were able to train a segmentation network on 2D MRI slices of brain tumor patients. The segmentation accuracy which we were able to attain is lower than the one achieved using supervised segmentation, but still in a close range. More importantly, in a clinical setting, most data is labeled on the image level. We hence believe that this segmentation method is of immense value for any segmentation task, where no data with manual segmentations are available.

Finally, we developed a small tool to analyze lesion developments on the basis of manually created lesion maps of a longitudinal multiple sclerosis lesion study. The le-

¹The code for MD-GRU is available at <https://github.com/zubata88/mdgru>.

sion maps were manually drawn on each brain scan. A coregistration of all present volumes allowed for the transformation of all lesion maps into the same reference system, which enabled us to analyse individual lesion developments over time.

Future Work

Provided there is enough training data, supervised segmentation using MD-GRU comes very close to human performance. Furthermore, the sometimes prevalent high inter-rater or even intra-rater variability suggests, that it is nearly impossible to exactly match manual segmentations, as there are usually ambiguous regions present, which do not clearly correspond to one of the classes. Instead of further trying to improve the accuracy of our method, we have an ongoing project which tries to simplify it. We try to replace the gated recurrent units in our segmentation method with orthogonally restricted simple hyperbolic tangent units. By not using any gates, they use a fraction of the computation time and consume less memory than a gated recurrent unit. We hope for faster training of the method, as the number of parameters is approximately a third compared to the current approach.

While the results of supervised segmentation methods are usually very convincing, the amount of manual labor to create the training data for such an approach is also overwhelming. We hence intend to focus on our latest weakly-supervised project, which learns to segment differences in the data of patients suffering from a specific disease and a healthy reference. The approach is so far just a proof of concept. Although reasonable segmentations could be generated, the other parts of the network did not always function as intended. Furthermore, there should be a way of drastically simplifying the loss function, as the high number of terms makes a hyper parameter search very cumbersome. Another point is the network formulation itself. Using only one generator and one discriminator should already suffice to create segmentations, as we do not necessarily care for realistic inpaintings. This would reduce the loss term to a very simple formulation, consisting only of a ℓ^1 -norm and a GAN term.

Conclusion

In this thesis, we were able to match and surpass the state of the art in supervised lesion segmentation and pioneered in the field of weakly-supervised lesion segmentation. Our MD-GRU has been adopted at different locations and is used for further, applied research. We see a lot of promising applications in the field of weakly-supervised segmentation and will focus on this interesting body of research. To allow for reproducible research and prevent reinventions of the wheel, the code to all our projects is going to be released.

Bibliography

- [1] Z. Akkus, A. Galimzianova, A. Hoogi, D. L. Rubin, and B. J. Erickson. Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions. *Journal of Digital Imaging*, 30(4):449–459, Aug. 2017.
- [2] B. Alfano, A. Brunetti, M. Larobina, M. Quarantelli, E. Tedeschi, A. Ciarmiello, E. M. Covelli, and M. Salvatore. Automated segmentation and measurement of global white matter lesion volume in patients with multiple sclerosis. *Journal of Magnetic Resonance Imaging*, 12(6):799–807, Dec. 2000.
- [3] P. Anbeek, K. L. Vincken, F. Groenendaal, A. Koeman, M. J. P. van Osch, and J. Van der Grond. Probabilistic Brain Tissue Segmentation in Neonatal Magnetic Resonance Imaging. *Pediatric Research*, 63(2):158–163, Feb. 2008.
- [4] P. Anbeek, K. L. Vincken, M. J. P. van Osch, R. H. C. Bisschops, and J. van der Grond. Probabilistic segmentation of white matter lesions in MR imaging. *NeuroImage*, 21(3):1037–1044, Mar. 2004.
- [5] P. Anbeek, K. L. Vincken, and M. A. Viergever. Automated MS-lesion segmentation by k-nearest neighbor classification. In *The MIDAS Journal-MS Lesion Segmentation (MICCAI Workshop)*, pages 1–8, 2008.
- [6] S. Andermatt, S. Pezold, M. Amann, and P. C. Cattin. Multi-dimensional gated recurrent units for automated anatomical landmark localization. *arXiv preprint arXiv:1708.02766*, 2017.
- [7] S. Andermatt, S. Pezold, and P. Cattin. Multi-dimensional Gated Recurrent Units for the Segmentation of Biomedical 3D-Data. In *Deep Learning and Data Labeling for Medical Applications*, pages 142–151. Springer, 2016.
- [8] M. Arjovsky, A. Shah, and Y. Bengio. Unitary Evolution Recurrent Neural Networks. *arXiv:1511.06464 [cs, stat]*, Nov. 2015.
- [9] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [10] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos. Segmentation Labels for the Pre-operative

- Scans of the TCGA-GBM collection. *The Cancer Imaging Archive*, 2017. DOI: 10.7937/K9/TCIA.2017.KLXWJJ1Q.
- [11] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos. Segmentation Labels for the Pre-operative Scans of the TCGA-LGG collection. *The Cancer Imaging Archive*, 2017. DOI: 10.7937/K9/TCIA.2017.GJQ7R0EF.
 - [12] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, J. B. Freymann, K. Farahani, and C. Davatzikos. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 4:170117, 2017.
 - [13] S. Bakas, B. Menze, M. Reyes, K. Farahani, J. Freymann, J. S. Kirby, et al. *Pre-conference Proceedings of the International Multimodal Brain Tumor Segmentation (BraTS) Challenge 2017*. 2017.
 - [14] R. Bakshi, S. Ariyaratana, R. H. B. Benedict, and L. Jacobs. Fluid-Attenuated Inversion Recovery Magnetic Resonance Imaging Detects Cortical and Juxtacortical Multiple Sclerosis Lesions. *Archives of Neurology*, 58(5):742–748, May 2001.
 - [15] A. Birenbaum and H. Greenspan. Longitudinal Multiple Sclerosis Lesion Segmentation Using Multi-view Convolutional Neural Networks. In *Deep Learning and Data Labeling for Medical Applications*, Lecture Notes in Computer Science, pages 58–67. Springer, Oct. 2016.
 - [16] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.
 - [17] F. Bloch. Nuclear induction. *Physical review*, 70(7-8):460, 1946.
 - [18] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.
 - [19] M. Brant-Zawadzki, G. D. Gillan, and W. R. Nitz. MP RAGE: A three-dimensional, T1-weighted, gradient-echo sequence—initial experience in the brain. *Radiology*, 182(3):769–775, Mar. 1992.
 - [20] T. Brosch, Y. Yoo, L. Y. W. Tang, D. K. B. Li, A. Traboulsee, and R. Tam. Deep convolutional encoder networks for multiple sclerosis lesion segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – (MICCAI)*, volume 9351 of *Lecture Notes in Computer Science*, pages 3–11. Springer International Publishing, 2015.
 - [21] M. E. Caligiuri, P. Perrotta, A. Augimeri, F. Rocca, A. Quattrone, and A. Cherubini. Automatic Detection of White Matter Hyperintensities in Healthy Aging

- and Pathology Using Magnetic Resonance Imaging: A Review. *Neuroinformatics*, 13(3):261–276, July 2015.
- [22] A. Carass, S. Roy, A. Jog, J. L. Cuzzocreo, E. Magrath, A. Gherman, J. Button, J. Nguyen, F. Prados, C. H. Sudre, M. Jorge Cardoso, N. Cawley, O. Ciccarelli, C. A. M. Wheeler-Kingshott, S. Ourselin, L. Catanese, H. Deshpande, P. Maurel, O. Commowick, C. Barillot, X. Tomas-Fernandez, S. K. Warfield, S. Vaidya, A. Chunduru, R. Muthuganapathy, G. Krishnamurthi, A. Jesson, T. Arbel, O. Maier, H. Handels, L. O. Ithme, D. Unay, S. Jain, D. M. Sima, D. Smeets, M. Ghafoorian, B. Platel, A. Birenbaum, H. Greenspan, P.-L. Bazin, P. A. Calabresi, C. M. Crainiceanu, L. M. Ellingsen, D. S. Reich, J. L. Prince, and D. L. Pham. Longitudinal multiple sclerosis lesion segmentation: Resource and challenge. *NeuroImage*, 148:77–102, Mar. 2017.
- [23] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.
- [24] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. The Loss Surfaces of Multilayer Networks. *arXiv:1412.0233 [cs]*, Nov. 2014.
- [25] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv:1412.3555 [cs]*, Dec. 2014.
- [26] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, editors, *Medical Image Computing and Computer-Assisted Intervention – (MICCAI)*, volume 9901 of *Lecture Notes in Computer Science*, pages 424–432. Springer International Publishing, 2016.
- [27] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, 22(12):3207–3220, Dec. 2010.
- [28] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *arXiv:1511.07289 [cs]*, Nov. 2015.
- [29] B. D. Coene, J. V. Hajnal, P. Gatehouse, D. B. Longmore, S. J. White, A. Oatridge, J. M. Pennock, I. R. Young, and G. M. Bydder. MR of the brain using fluid-attenuated inversion recovery (FLAIR) pulse sequences. *American Journal of Neuroradiology*, 13(6):1555–1564, Nov. 1992.
- [30] D. Crevier. *AI: the tumultuous history of the search for artificial intelligence*, page 203. Basic Books, 1993.

- [31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.
- [32] M. Drozdal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal. The Importance of Skip Connections in Biomedical Image Segmentation. In *Deep Learning and Data Labeling for Medical Applications*, Lecture Notes in Computer Science, pages 179–187. Springer, Oct. 2016.
- [33] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2012.
- [34] K. Farahani, B. Menze, and M. Reyes. *Brats 2014 Challenge Manuscripts (2014)*. 2014.
- [35] R. Felix, W. Schörner, M. Laniado, H. P. Niendorf, C. Claussen, W. Fiegler, and U. Speck. Brain tumors: MR imaging with gadolinium-DTPA. *Radiology*, 156(3):681–688, Sept. 1985.
- [36] M. Filippi, F. Barkhof, S. Bressi, T. A. Yousry, and D. H. Miller. Inter-rater variability in reporting enhancing lesions present on standard and triple dose gadolinium scans of patients with multiple sclerosis. *Multiple Sclerosis (Houndmills, Basingstoke, England)*, 3(4):226–230, Aug. 1997.
- [37] M. Filippi, M. A. Horsfield, S. Bressi, V. Martinelli, C. Baratti, P. Reganati, A. Campi, D. H. Miller, and G. Comi. Intra- and inter-observer agreement of brain MRI lesion volume measurements in multiple sclerosis. *Brain*, 118(6):1593–1600, Dec. 1995.
- [38] D. García-Lorenzo, S. Francis, S. Narayanan, D. L. Arnold, and D. L. Collins. Review of automatic segmentation methods of multiple sclerosis white matter lesions on conventional magnetic resonance imaging. *Medical Image Analysis*, 17(1):1–18, Jan. 2013.
- [39] Y. Ge. Multiple Sclerosis: The Role of MR Imaging. *American Journal of Neuroradiology*, 27(6):1165–1176, June 2006.
- [40] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: continual prediction with lstm. *IET Conference Proceedings*, pages 850–855, January 1999.
- [41] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.
- [42] D. Goldberg-Zimring, A. Achiron, S. Miron, M. Faibel, and H. Azhari. Automated detection and characterization of multiple sclerosis lesions in brain MR images. *Magnetic resonance imaging*, 16(3):311–318, 1998.

- [43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [44] N. Gordillo, E. Montseny, and P. Sobrevilla. State of the art survey on MRI brain tumor segmentation. *Magnetic Resonance Imaging*, 31(8):1426–1438, Oct. 2013.
- [45] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947, 2000.
- [46] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.
- [47] M. Havaei, N. Guizard, H. Larochelle, and P.-M. Jodoin. Deep Learning Trends for Focal Brain Pathology Segmentation in MRI. In *Machine Learning for Health Informatics*, Lecture Notes in Computer Science, pages 125–148. Springer, 2016.
- [48] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
- [49] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [50] D. O. Hebb. *The organization of behavior: A neuropsychological theory*. Wiley, New York, June 1949.
- [51] G. E. Hinton. To recognize shapes, first learn to generate images. *Progress in brain research*, 165:535–547, 2007.
- [52] G. E. Hinton and R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, July 2006.
- [53] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580 [cs]*, July 2012.
- [54] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [55] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*, pages 286–297. Springer, 1990.
- [56] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat’s striate cortex. *The Journal of Physiology*, 148(3):574–591, Oct. 1959.

- [57] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]*, Feb. 2015.
- [58] S. Jain, A. Ribbens, D. M. Sima, S. V. Huffel, F. Maes, and D. Smeets. Unsupervised Framework for Consistent Longitudinal MS Lesion Segmentation. In *Medical Computer Vision and Bayesian and Graphical Models for Biomedical Imaging*, Lecture Notes in Computer Science, pages 208–219. Springer, Oct. 2016.
- [59] S. Jain, D. M. Sima, A. Ribbens, M. Cambron, A. Maertens, W. Van Hecke, J. De Mey, F. Barkhof, M. D. Steenwijk, M. Daams, F. Maes, S. Van Huffel, H. Vrenken, and D. Smeets. Automatic segmentation and volumetry of multiple sclerosis brain lesions from MR images. *NeuroImage. Clinical*, 8:367–375, 2015.
- [60] R. Jozefowicz, W. Zaremba, and I. Sutskever. An Empirical Exploration of Recurrent Network Architectures. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 2342–2350, 2015.
- [61] J. Juan-Albarracín, E. Fuster-Garcia, J. V. Manjón, M. Robles, F. Aparici, L. Martí-Bonmatí, and J. M. García-Gómez. Automated glioblastoma segmentation based on a multiparametric structured unsupervised classification. *PloS one*, 10(5):e0125143, 2015.
- [62] M. Kamber, R. Shinghal, D. Collins, G. Francis, and A. Evans. Model-based 3-D segmentation of multiple sclerosis lesions in magnetic resonance brain images. *IEEE Transactions on Medical Imaging*, 14(3):442–453, Sept. 1995.
- [63] K. Kamnitsas, W. Bai, E. Ferrante, S. G. McDonagh, M. Sinclair, N. Pawlowski, M. Rajchl, M. C. H. Lee, B. Kainz, D. Rueckert, and B. Glocker. Ensembles of multiple models and architectures for robust brain tumour segmentation. *CoRR*, abs/1711.01468, 2017.
- [64] K. Kamnitsas, C. Baumgartner, C. Ledig, V. Newcombe, J. Simpson, A. Kane, D. Menon, A. Nori, A. Criminisi, D. Rueckert, and B. Glocker. Unsupervised Domain Adaptation in Brain Lesion Segmentation with Adversarial Networks. In *Information Processing in Medical Imaging*, Lecture Notes in Computer Science, pages 597–609. Springer, June 2017.
- [65] K. Kamnitsas, E. Ferrante, S. Parisot, C. Ledig, A. V. Nori, A. Criminisi, D. Rueckert, and B. Glocker. DeepMedic for brain tumor segmentation. In *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 138–149. Springer, 2016.
- [66] D. Katz, J. K. Taubenberger, B. Cannella, D. E. McFarlin, C. S. Raine, and H. F. McFarland. Correlation between magnetic resonance imaging findings and lesion development in chronic, active multiple sclerosis. *Annals of Neurology*, 34(5):661–669, Nov. 1993.

- [67] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [68] D. P. Kingma and M. Welling. Stochastic gradient VB and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*, 2014.
- [69] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter. Self-Normalizing Neural Networks. *arXiv:1706.02515 [cs, stat]*, June 2017.
- [70] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [71] H. B. W. Larsson, M. Stubgaard, J. L. Frederiksen, M. Jensen, O. Henriksen, and O. B. Paulson. Quantitation of blood-brain barrier defect by magnetic resonance imaging and gadolinium-DTPA in patients with multiple sclerosis and brain tumors. *Magnetic Resonance in Medicine*, 16(1):117–131, Oct. 1990.
- [72] J. Lecoœur, S. P. Morrissey, J.-C. Ferré, D. L. Arnold, D. L. Collins, and C. Barillot. Multiple Sclerosis Lesions Segmentation using Spectral Gradient and Graph Cuts. In *Medical Image Analysis on Multiple Sclerosis (validation and methodological issues)*, pages 92–103, New York City, United States, Sept. 2008.
- [73] Y. Lecun. Une procedure d’apprentissage pour reseau a seuil asymmetrique (a learning scheme for asymmetric threshold networks). In *Proceedings of Cognitiva 85, Paris, France*, pages 599–604, 1985.
- [74] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [75] H. Li, R. Zhao, and X. Wang. Highly Efficient Forward and Backward Propagation of Convolutional Neural Networks for Pixelwise Classification. *arXiv:1412.4526 [cs]*, Dec. 2014.
- [76] X. Lladó, A. Oliver, M. Cabezas, J. Freixenet, J. C. Vilanova, A. Quiles, L. Valls, L. Ramió-Torrentà, and À. Rovira. Segmentation of multiple sclerosis lesions in brain MRI: A review of automated approaches. *Information Sciences*, 186(1):164–185, Mar. 2012.
- [77] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [78] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

- [79] O. Maier, B. H. Menze, J. von der Gablentz, L. Häni, M. P. Heinrich, M. Liebrand, S. Winzeck, A. Basit, P. Bentley, L. Chen, D. Christiaens, F. Dutil, K. Egger, C. Feng, B. Glocker, M. Götz, T. Haeck, H.-L. Halme, M. Havaei, K. M. Iftekharuddin, P.-M. Jodoin, K. Kamnitsas, E. Kellner, A. Korvenoja, H. Larochelle, C. Ledig, J.-H. Lee, F. Maes, Q. Mahmood, K. H. Maier-Hein, R. McKinley, J. Muschelli, C. Pal, L. Pei, J. R. Rangarajan, S. M. S. Reza, D. Robben, D. Rueckert, E. Salli, P. Suetens, C.-W. Wang, M. Wilms, J. S. Kirschke, U. M. Krämer, T. F. Munte, P. Schramm, R. Wiest, H. Handels, and M. Reyes. ISLES 2015 - A public evaluation benchmark for ischemic stroke lesion segmentation from multispectral MRI. *Medical Image Analysis*, 35:250–269, Jan. 2017.
- [80] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, Dec. 1943.
- [81] W. I. McDonald, A. Compston, G. Edan, D. Goodkin, H.-P. Hartung, F. D. Lublin, H. F. McFarland, D. W. Paty, C. H. Polman, S. C. Reingold, M. Sandberg-Wollheim, W. Sibley, A. Thompson, S. Van Den Noort, B. Y. Weinshenker, and J. S. Wolinsky. Recommended diagnostic criteria for multiple sclerosis: Guidelines from the international panel on the diagnosis of multiple sclerosis. *Annals of Neurology*, 50(1):121–127, July 2001.
- [82] H. F. McFarland and R. Martin. Multiple sclerosis: a complicated picture of autoimmunity. *Nature immunology*, 8(9):913, 2007.
- [83] A. M. Mendrik, K. L. Vincken, H. J. Kuijf, M. Breeuwer, W. H. Bouvy, J. de Bresser, A. Alansary, M. de Bruijne, A. Carass, A. El-Baz, A. Jog, R. Katyal, A. R. Khan, F. van der Lijn, Q. Mahmood, R. Mukherjee, A. van Oproek, S. Paneri, S. Pereira, M. Persson, M. Rajchl, D. Sarikaya, Ö. Smedby, C. A. Silva, H. A. Vrooman, S. Vyas, C. Wang, L. Zhao, G. J. Biessels, and M. A. Viergever. MRBrainS Challenge: Online Evaluation Framework for Brain Image Segmentation in 3T MRI Scans. <https://www.hindawi.com/journals/cin/2015/813696/>, 2015.
- [84] B. Menze, M. Reyes, K. Farahani, J. Kalpathy-Cramer, and D. Kwon. *Brats 2015 Challenge Manuscripts (2015)*. 2015.
- [85] B. Menze, M. Reyes, J. Kalpathy-Cramer, K. Farahani, and S. Bakas. *Brats 2016 Challenge Manuscripts (2016)*. 2016.
- [86] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M. A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, Ç. Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan,

- D. Sarikaya, L. Schwartz, H. C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. V. Leemput. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, Oct. 2015.
- [87] D. H. Miller, F. Barkhof, and J. J. P. Nauta. Gadolinium enhancement increases the sensitivity of MRI in detecting disease activity in multiple sclerosis. *Brain*, 116(5):1077–1094, Oct. 1993.
- [88] R. Milo and A. Miller. Revised diagnostic criteria of multiple sclerosis. *Autoimmunity reviews*, Jan. 2014.
- [89] M. L. Minsky and S. Papert. *Perceptrons: an introduction to computational geometry*. MIT press, 1969.
- [90] J. Morra, Z. Tu, A. Toga, and P. Thompson. Automatic segmentation of MS lesions using a contextual model for the MICCAI grand challenge. *Multiple Sclerosis Lesion Segmentation Challenge*, pages 1–7, 2008.
- [91] J. P. Mugler and J. R. Brookeman. Three-dimensional magnetization-prepared rapid gradient-echo imaging (3D MP RAGE). *Magnetic Resonance in Medicine*, 15(1):152–157, July 1990.
- [92] S. Pereira, A. Pinto, V. Alves, and C. A. Silva. Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE transactions on medical imaging*, 35(5):1240–1251, 2016.
- [93] C. H. Polman, S. C. Reingold, B. Banwell, M. Clanet, J. A. Cohen, M. Filippi, K. Fujihara, E. Havrdova, M. Hutchinson, L. Kappos, F. D. Lublin, X. Montalban, P. O’Connor, M. Sandberg-Wollheim, A. J. Thompson, E. Waubant, B. Weinshenker, and J. S. Wolinsky. Diagnostic criteria for multiple sclerosis: 2010 Revisions to the McDonald criteria. *Annals of Neurology*, 69(2):292–302, 2011.
- [94] C. H. Polman, S. C. Reingold, G. Edan, M. Filippi, H.-P. Hartung, L. Kappos, F. D. Lublin, L. M. Metz, H. F. McFarland, P. W. O’Connor, M. Sandberg-Wollheim, A. J. Thompson, B. G. Weinshenker, and J. S. Wolinsky. Diagnostic criteria for multiple sclerosis: 2005 revisions to the “McDonald Criteria”. *Annals of Neurology*, 58(6):840–846, Dec. 2005.
- [95] R. Raina, A. Madhavan, and A. Y. Ng. Large-scale Deep Unsupervised Learning Using Graphics Processors. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML*, pages 873–880, New York, NY, USA, 2009. ACM.
- [96] I. Rekik, S. Allasonnière, T. K. Carpenter, and J. M. Wardlaw. Medical image analysis methods in MR/CT-imaged acute-subacute ischemic stroke lesion: Segmentation, prediction and insights into dynamic evolution simulation models. A critical appraisal. *NeuroImage: Clinical*, 1(1):164–178, Jan. 2012.

- [97] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [98] F. Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.
- [99] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [100] F. Rosenblatt. *Principles of neurodynamics*. Spartan Book, 1962.
- [101] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, Oct. 1986.
- [102] B. R. Sajja, S. Datta, R. He, M. Mehta, R. K. Gupta, J. S. Wolinsky, and P. A. Narayana. Unified Approach for Multiple Sclerosis Lesion Segmentation on Brain MRI. *Annals of Biomedical Engineering*, 34(1):142–151, Mar. 2006.
- [103] N. Sauwen, M. Acou, S. Van Cauter, D. Sima, J. Veraart, F. Maes, U. Himmelreich, E. Achten, and S. Van Huffel. Comparison of unsupervised classification methods for brain tumor segmentation using multi-parametric MRI. *NeuroImage : Clinical*, 12:753–764, Sept. 2016.
- [104] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv:1312.6120 [cond-mat, q-bio, stat]*, Dec. 2013.
- [105] H. G. Schmidt, G. R. Norman, and H. P. Boshuizen. A cognitive perspective on medical expertise: Theory and implication. *Academic Medicine: Journal of the Association of American Medical Colleges*, 65(10):611–621, Oct. 1990.
- [106] M. Scully, V. Magnotta, C. Gasparovic, P. Pelligrino, D. Feis, and H. Bockholt. 3d segmentation in the clinic: a grand challenge ii at miccai 2008–ms lesion segmentation. *Multiple Sclerosis Lesion Segmentation Challenge*, pages 1–9, 2008.
- [107] N. Shiee, P.-L. Bazin, J. L. Cuzzocreo, D. S. Reich, P. A. Calabresi, and D. L. Pham. Topologically constrained segmentation of brain images with multiple sclerosis lesions. *Medical Image Analysis on Multiple Sclerosis (MICCAI Workshop)*, pages 71–81, 2008.
- [108] N. Shiee, P.-L. Bazin, A. Ozturk, D. S. Reich, P. A. Calabresi, and D. L. Pham. A topology-preserving approach to the segmentation of brain images with multiple sclerosis lesions. *NeuroImage*, 49(2):1524–1535, Jan. 2010.
- [109] N. Shiee, P.-L. Bazin, and D. Pham. Multiple sclerosis lesion segmentation using statistical and topological atlases. In *Grand Challenge Workshop: Multiple Sclerosis Lesion Segmentation Challenge*, pages 1–10, 2008.

- [110] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, Sept. 2014.
- [111] J.-C. Souplet, C. Lebrun, N. Ayache, G. Malandain, and others. An automatic segmentation of T2-FLAIR multiple sclerosis lesions. In *The MIDAS Journal-MS Lesion Segmentation (MICCAI Workshop)*, 2008.
- [112] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [113] L. Steinman. Multiple sclerosis: A two-stage disease. *Nature Immunology*, 2(9):762–764, Sept. 2001.
- [114] M. F. Stollenga, W. Byeon, M. Liwicki, and J. Schmidhuber. Parallel Multi-Dimensional LSTM, With Application to Fast Biomedical Volumetric Image Segmentation. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2998–3006. Curran Associates, Inc., 2015.
- [115] N. K. Subbanna, S. J. Francis, D. Precup, D. L. Collins, D. L. Arnold, and T. Arbel. Adapted mrf segmentation of multiple sclerosis lesions using local contextual information. In *Proceedings of Medical Image Understanding and Analysis (MIUA)*, pages 351–356, 2011.
- [116] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [117] A. A. Taha and A. Hanbury. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Medical Imaging*, 15:29, Aug. 2015.
- [118] T. Tieleman and G. Hinton. Lecture 6.5-Rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 2012.
- [119] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Improved Texture Networks: Maximizing Quality and Diversity in Feed-forward Stylization and Texture Synthesis. *arXiv:1701.02096 [cs]*, Jan. 2017.
- [120] S. Valverde, M. Cabezas, E. Roura, S. González-Villà, D. Pareto, J.-C. Vilanova, L. Ramió-Torrentà, À. Rovira, A. Oliver, and X. Lladó. Improving automated multiple sclerosis lesion segmentation with a cascaded 3D convolutional neural network approach. *arXiv:1702.04869 [cs]*, Feb. 2017.
- [121] K. Van Leemput, F. Maes, D. Vandermeulen, A. Colchester, and P. Suetens. Automated segmentation of multiple sclerosis lesions by model outlier detection. *IEEE Transactions on Medical Imaging*, 20(8):677–688, Aug. 2001.

- [122] G. Vishnuvarthanan, M. P. Rajasekaran, P. Subbaraj, and A. Vishnuvarthanan. An unsupervised learning method with a clustering approach for tumor identification and tissue segmentation in magnetic resonance brain images. *Applied Soft Computing*, 38:190–212, Jan. 2016.
- [123] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pages 1058–1066, 2013.
- [124] N. Weiss, D. Rueckert, and A. Rao. Multiple sclerosis lesion segmentation using dictionary learning and sparse coding. In K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, editors, *Medical Image Computing and Computer-Assisted Intervention – (MICCAI)*, volume 8149 of *Lecture Notes in Computer Science*, pages 735–742. Springer, 2013.
- [125] H. Wu. Global stability analysis of a general class of discontinuous neural networks with linear growth activation functions. *Information Sciences*, 179(19):3432–3441, Sept. 2009.
- [126] Y. Wu, S. K. Warfield, I. L. Tan, W. M. Wells III, D. S. Meier, R. A. van Schijndel, F. Barkhof, and C. R. G. Guttmann. Automated segmentation of multiple sclerosis lesion subtypes with multichannel MRI. *NeuroImage*, 32(3):1205–1215, Sept. 2006.
- [127] R. M. Young, A. Jamshidi, G. Davis, and J. H. Sherman. Current trends in the surgical management and treatment of adult glioblastoma. *Annals of Translational Medicine*, 3(9), June 2015.
- [128] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.
- [129] M. D. Zeiler. ADADELTA: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.
- [130] M. D. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. *arXiv:1311.2901 [cs]*, Nov. 2013.
- [131] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [132] A. P. Zijdenbos, R. Forghani, and A. C. Evans. Automatic “pipeline” analysis of 3-d MRI data for clinical trials: application to multiple sclerosis. *IEEE transactions on medical imaging*, 21(10):1280–1291, 2002.

Publications

Peer-reviewed Publications

S. Andermatt, S. Pezold, and P. Cattin. Automated Segmentation of Multiple Sclerosis Lesions using Multi-Dimensional Gated Recurrent Units. In *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Springer, 2017

S. Andermatt, A. Papadopoulou, E.-W. Radue, T. Sprenger, and P. Cattin. Tracking the evolution of cerebral gadolinium-enhancing lesions to persistent t1 black holes in multiple sclerosis: Validation of a semiautomated pipeline. *Journal of Neuroimaging*, 27(5):469–475, 2017

S. Andermatt, S. Pezold, and P. Cattin. Multi-dimensional Gated Recurrent Units for the Segmentation of Biomedical 3D-Data. In *International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*, pages 142–151. Springer, 2016

Technical Reports

S. Andermatt, S. Pezold, M. Amann, and P. C. Cattin. Multi-dimensional gated recurrent units for automated anatomical landmark localization. *arXiv preprint arXiv:1708.02766*, 2017

S. Andermatt, S. Pezold, and P. Cattin. Pathology Segmentation using Distributional Differences to Images of Healthy Origin. *arXiv preprint arXiv:1805.10344*, 2018

R. Sandkühler, C. Jud, **S. Andermatt**, and P. Cattin. AirLab: Autograd Image Registration Laboratory. *arXiv preprint arXiv:1806.09907*, 2018

Posters

O. Pusterla, **S. Andermatt**, G. Baumann, S. Nyilas, P. Madörin, T. Haas, S. Pezold, F. Santini, P. Latzin, P. Cattin, and O. Bieri. Deep Learning Lung Segmentation in

Paediatric Patients. In *Proceedings of the 26th Annual Meeting of the ISMRM*, Paris, France, 2018

S. Andermatt, S. Pezold, and P. Cattin. Multi-Dimensional GRU for the Segmentation of White Matter Hyperintensities, *WMH Challenge*, MICCAI 2017

S. Andermatt, S. Pezold, and P. Cattin. Multi-Dimensional GRU for Brain Tumor Segmentation, *BraTS Challenge*, MICCAI 2017

S. Andermatt, S. Pezold, and P. Cattin. Automatic Segmentation of Brain Pathology using Recurrent Neural Networks, *DBE Research Day*, 2017

S. Andermatt, S. Pezold, and P. Cattin. Automatic Segmentation of Multiple Sclerosis Lesions using Recurrent Neural Networks, *DBE Research Day*, 2016

S. Andermatt, S. Pezold, and P. Cattin. Automatic Detection and Segmentation of Multiple Sclerosis Lesions in MRI, *DBE Research Day*, 2015

Talks

Automated Segmentation of Multiple Sclerosis Lesions using Multi-Dimensional Gated Recurrent Units, *BRAINLES Satellite Event MICCAI 2017*, Quebec City, 2017

Multi-Dimensional GRU for the Segmentation of Biomedical Data, *DLMIA Satellite Event MICCAI 2016*, Athens, 2016

Automatic Segmentation of Brain Structures using Multi-Dimensional Recurrent Neural Networks, *DBE Research Day*, Basel, 2016

Segmentation of Focal Lesions in the Brain, *Special Interest Group*, Allschwil, 2016